Amendments to the Specification:

Please add the following new paragraph beginning at page 1, after line 1 and before line 2:

Computer Appendix

This application includes a Computer Program Listing Appendix on compact disc, hereby incorporated by reference.

Please replace the paragraph beginning at page 1, line 6 through page 2, 13 with the following amended paragraph:

This case is also related to the following copending applications, all filed on October 16, 1995: REMOTE CONTROL INTERFACE, by B.R. Banerjee, S.C. Gladwin, A. Maskatia and A. Soucy, Serial No. 08/543,700; RADIO FLASH UPDATE, by D. Bi, H. Hsiung and J. Wilson, Serial No. 08/543,463; MOUSE EMULATION WITH PASSIVE PEN, by D. Bi, G. Cohen, M. Cortopassi, J. George, S.C. Gladwin, H. Hsiung, P. Lim, J. Parham, A. Soucy, D. Voegeli and J. Wilson, Serial No. 08/543,786; RESUME ON PEN CONTACT, by M. Cortopassi, S.C. Gladwin and D. Voegeli, Serial No. 08/543,510; SCREEN SAVER DISABLER, by D. Bi, S.C. Gladwin and J. Wilson, Serial No. 08/543,698; IPX DRIVER FOR MULTIPLE LAN ADAPTERS, by D. Bi, Serial No. 08/553,808; DISASTER RECOVERY JUMPER, by M. Cortopassi, J. George, J. Parham and D. Voegeli, Serial No. 08/543,423; RC TIME CONSTANT, by M. Cortopassi, Serial No. 08/543,697; DOUBLE PEN UP EVENT, by D. Bi and J. George, Serial No. 08/543,787; REMOTE OCCLUSION REGION, by J. Wilson, Serial No. 08/543,701; BROADCAST SEARCH FOR AVAILABLE HOST, by D. Bi, S.C. Gladwin and J. Wilson, Serial No. 08/543,599; HOST/REMOTE CONTROL MODE, by M. Cortopassi, J. George, S.C. Gladwin, H. Hsiung, P. Lim, J. Parham, D. Voegeli and J. Wilson, Serial No. 08/551,936; PASSWORD SWITCH TO OVERRIDE REMOTE CONTROL, by D. Bi, S.C. Gladwin and J. Wilson, Serial No. 08/543,785; AUTOMATIC RECONNECT ON REQUIRED SIGNAL, by S.C. Gladwin and J. Wilson, Serial No. 08/543,425; and PORTABLE TABLET, by G. Cohen,

S.C. Gladwin, P. Lim, J. Smith, A. Soucy, K. Swen, G. Wong, K. Wood and G. Wu, Serial No. 29/045,319; REMOTE KEYBOARD MACROS ACTIVATED BY HOT ICONS, by S.C. Gladwin, 08/543,788, J. Wilson, Serial No. 08/543,788.

Please replace the paragraph beginning at page 2, line 14 through line 32 with the following amended paragraph:

This case is also related to the following cases, all filed on even date: MULTIPLE WIRELESS INTERFACES TO A SINGLE SERVER, by S.C. Gladwin, A. Soucy and J. Wilson, Serial No. 08/783,708; WIRELESS ENUMERATION OF AVAILABLE SERVERS, by S.C. Gladwin, D. Bi, A. Gopalan, and J. Wilson, Serial No. 08/784,275; DYNAMIC SERVER ALLOCATION FOR LOAD BALANCING WIRELESS INTERFACE PROCESSING, by D. Bi, Serial No. 08/784,276; DATA COMPRESSION LOADER, by D. Boals and J. Wilson, Serial No. 08/784,211; MULTI-USER RADIO FLASH ROM UPDATE, by D. Bi and J. Wilson, Serial No. 08/783,080; AUDIO COMPRESSION IN A WIRELESS INTERFACE DEVICE, by S.C. Gladwin, D. Bi and D. Voegeli, Serial No. 08/784,141; MULTI-USER ON-SCREEN KEYBOARD, by D. Bi, Serial No. 08/784,243; LOCAL HANDWRITING RECOGNITION IN A WIRELESS INTERFACE TABLET, by S. C. Gladwin, D. Bi, D. Boals and J. Wilson, Serial No. 08/784,034; INK TRAILS ON A WIRELESS REMOTE INTERFACE TABLET, by S.C. Gladwin, D. Bi, D. Boals, J. George, S. Merkle and J. Wilson, Serial No. 08/784,688, and MODE SWITCHING FOR PEN-BASED COMPUTER SYSTEMS, by D. Bi, Serial No. 08/784,212.

Please replace the paragraph beginning at page 5, line 14 through line 16 with the following amended paragraph:

FIGS. [[11-30]]11a-11f, 12a-12h, 13a-13d, 14a-14f, 15a-15g, 16a-16d, 17a-17f, 18a-18d, 19a-19r, 20a-20e, 21a-21i, 22a-22d, 23a-23e, 24a-24b, 25a-25c, 26a-26d, 27a-27c, 28a-28c, 29a-29g, 30a-30e are schematic diagrams of the wireless interface device in accordance with the present invention;

Please replace the paragraph beginning at page 6, line 11 through line 12 with the following amended paragraph:

~~FIG. 59 is a flow chart~~ FIGS. 59a and 59n are flow charts relating to a system for broadcasting for available hosts;

Please replace the paragraph beginning at page 6, line 16 through line 17 with the following amended paragraph:

FIGS. 62A-62C, and 63~~A and 63B~~ are flow charts relating to a wireless flash memory device programmer;

Please replace the paragraph beginning at page 6, line 18 through line 19 with the following amended paragraph:

FIGS. ~~64A and 64B~~ 64a-64c are flow charts relating to a system for providing automatic reconnection of the host;

Please replace the paragraph beginning at page 7, line 27 through line 29 with the following amended paragraph:

FIGs. [[80-85]]80a and 80b, and 81-85 are flow charts for enabling the FLASH memory device on multiple wireless interface devices to be updated wirelessly.

Please delete the paragraphs beginning at page 7, line 30 which starts with "FIGs. 86 and 87 are flow charts for an audio" through and including page 8, line 20 which ends with "present invention."

Please replace the paragraph beginning at page 8, line 16 through line 17 with the following amended paragraph:

FIGs. ~~98-109~~98-108 and 109a-109b represent flow charts for the invention illustrated in FIG. 97.

Please replace the paragraph beginning at page 15, line 14 through page 16, line 11 with the following amended paragraph:

In order to conserve battery power, the wireless interface device 100 incorporates power management. While a user of the wireless interface device 100 would normally only be aware of four power management states: "off"; "active"; "suspend"; and "sleep" modes, internally six power management states are implemented as shown in FIG. 5. More particularly, with reference to FIG. 5, before the wireless interface device 100 is powered up, the wireless interface device 100 is in an "off" state, indicated by the reference numeral 160. In an "off" state 160, no power is supplied to the system. A state 161 (the "active" state) is entered when the power switch (FIG. 28a) to the wireless interface device 100 is turned to the "on" position. In the active state 161, all components of wireless interface device 100 are active. From active state 161, the wireless interface device 100 enters a "local standby" state 162. The local standby state 162 is transparent to the user of the wireless interface device 100. From the user's point of view, in the local standby state 162, the wireless interface device 100 is in active mode. In this state 162, specific inactive devices are each put into a static state after a predetermined time-out period of inactivity for that device. In a static state, each device consumes minimal power. In the local standby state 162, devices that can be put into static states include the CPU 112, the video controller 113A, the pen controller 110A, the UART 134, and the transceiver 116. Backlighting of the LCD video display is also disabled in local standby state 162. If not, input activities are detected by the keyboard controller 125 or pen controller 110A. After the later of their respective present time-out periods, these devices are placed in a static state. These devices emerge from the static state once an activity relevant to its operation is detected, e.g. a pen event is detected.

Please replace the paragraph beginning at page 16, line 12 through page 17, line 2 with the following amended paragraph:

The user of the wireless interface device 100 can place the wireless interface device 100 in a "sleep" mode 163 by selecting an icon (FIG. [[37]]5) labelled "sleep" from the GUI as will be discussed below. Alternatively, the "sleep" mode may be entered from the active state 161

after a preset period of inactivity. In a "sleep" mode, corresponding to either "sleep" state 163 or "active sleep" state 164, the display subsystem 113 is switched off; and most devices are placed in static states. When a keyboard or pen event is detected, the sleep state 163 and active sleep state 164 are exited, and the wireless interface device 100 enters the active state 161. From the sleep state 163, an active sleep state 164 is entered when a communication packet is received from the host computer 101. Although the display subsystem 113 is turned off, the received communication packet can result in an update to an image stored in the video memory 113B. The CPU 112 handles the communication packet from the host computer 101 and activates the video controller 113A to update such an image. The active sleep state 164 is transparent to the user of the wireless interface device 100, since the updated image is not displayed on the LCD screen 113C. When the communication packet is handled, the wireless interface 100 returns to a sleep state 163. The device activities in wireless interface device 100 in "sleep" mode 163 are illustrated in Table 1 below.

Please replace the paragraph beginning at page 30, line 28 through page 31, line 14 with the following amended paragraph:

One embodiment of the invention is illustrated in the schematic drawings, FIGS. [[11-30]]11a-11f, 12a-12h, 13a-13d, 14a-14f, 15a-15g, 16a-16d, 17a-17f, 18a-18d, 19a-19r, 20a-20e, 21a-21i, 22a-22d, 23a-23e, 24a-24b, 25a-25c, 26a-26d, 27a-27c, 28a-28c, 29a-29g, 30a-30e. Referring to FIG. 11a, the system may include a CPU 112, such as an AMD Model No. AM386DXLV microprocessor. The CPU 112 includes a 32-bit data bus D[0..31] as well as a 32-bit address bus A[2..31]. Both the data bus D[0..31] as well as the address bus A[2..31] are connected to the processor bus 150 (FIG. 4), for example, an AT bus. As will be discussed in more detail below, the system controller 129 (FIG. 4) performs various functions including management of the processor bus 150. In order to conserve power, a 3-volt microprocessor may be used for the CPU 112. As such, a 3-volt supply 3V_CPU is applied to the power supply VCC pins on the CPU 112. The 3-volt supply 3V_CPU is available from a DC-to-DC converter 300

(FIG. 26a) by way of a ferrite bead inductor 302. In particular, the DC-to-DC converter 300 includes a 3-volt output, 3V_CORE. This output, 3V_CORE, is applied to the ferrite bead inductor 302 and, in turn, to the power supply pins VCC of the CPU 112. In order to prevent noise and fluctuations in the power supply voltage from affecting the operation of the CPU 112, the power supply voltage 3V_CPU is filtered by a plurality of bypass capacitors 304 through 330.

Please replace the paragraph beginning at page 31, line 22 through line 29 with the following amended paragraph:

The CPU 112 is adapted to operate at 25 megahertz (MHz) at 3.0 volts. A 25 MHz clock signal, identified as CPU CLK, available from a clock generator 398 (FIG. 13d), is applied to a clock input CLK2 on the CPU 112 by way of a resistor 349 and a pair of capacitors 351 and 353. The AMD Model No. AMD386DXLV microprocessor supports a static state, which enables the clock to be halted and restarted at any time.

Please replace the paragraph beginning at page 31, line 30 through page 32, line 13 with the following amended paragraph:

The wireless interface device 100 includes a speaker 355. The speaker 355 is under the control of the system controller 129 (FIG. 12g). In particular, a speaker control signal SPKR from the system controller 129 is applied to a source terminal of a field-effect transistor (FET) 357 for direct control of the speaker 355. The drain terminal is connected to the speaker 355 by way of a current-limiting resistor 359 and a bypass capacitor 371. Normally, the speaker 355 is active all the time. In particular, the gate terminal of the FET 357 is connected to the system ground by way of a resistor 373. The gate terminal of the FET 357 is also under the control of a speaker disable signal SPKRDISABLE, available from the keyboard controller 125 (FIG. 15a). The speaker disable signal SPKRDISABLE is active high. Thus, when the speaker disable signal SPKRDISABLE signal is low, the FET 357 is turned on to enable the speaker signal SPKR from the system controller 129 to control the speaker 355. When the speaker disable signal SPKRDISABLE is high, the FET 357 is turned off to disable the speaker 355.

Please replace the paragraph beginning at page 32, line 14 through line 25 with the following amended paragraph:

Referring to FIG. 12a-12h, the system controller 129 is connected between the local processor or AT bus 150 and the system ISA bus 151. The system controller 129 performs a variety of functions including that of system controller, DRAM controller, power management, battery management and management of the local AT bus 150. The system controller 129, preferably a PicoPower Pine Evergreen 3, Model No. 86C368 system controller, is a 208-pin device that operates at 33 MHz with a full 5-volt input or a hybrid 5-volt/3.3-volt input. At 3.3 volts the system controller 129 is adapted to reliably operate at 20 Mhz and perhaps up to 25 Mhz.

Please replace the paragraph beginning at page 32, line 33 through page 33, line 7 with the following amended paragraph:

The system controller 129 supports both fast GATE A20 and a fast reset control of the CPU 112. In particular, the system controller 129 includes a 32-bit address bus A[0..31] that is connected to the local AT bus 150. The address line A[20] is used to develop a signal CPUA20, which is applied to the A20 pin on the CPU 112 and also applied to an AND gate 379 (FIG. 11b) to support a port 92H for a fast GATE A20 signal. A fast reset signal RSTCPU is also generated by the system controller 129. The fast reset signal RSTCPU is applied to the reset pin RESET of the CPU 112 for fast reset control.

Please replace the paragraph beginning at page 33, line 8 through line 19 with the following amended paragraph:

The system controller 129 also provides various other system level functions. For example, the system controller 129 includes a register at address 300H. By setting bit 12 of this register, a ROM chip select signal ROMCS is generated, which enables writes to the flash memory system 117 (FIG. 25c), which will be discussed below. A keyboard controller chip select signal KBDCS for the keyboard controller 125 (FIG. 15a), as well as general purpose chip

select signals GPCS1 and GPCS2 for selecting between the RF controller 114A, the UART 134 (FIG. 16a) or the pen controller 110A (FIG. 21e), are generated by the system controller 129.

Please replace the paragraph beginning at page 33, line 27 through page 34, line 4 with the following amended paragraph:

All of the ground pins GND on the system controller 129 are tied to the system ground. Both 3-volt and 5-volt power supplies are applied to the system controller 129. In particular, a 5-volt supply 5V_EG is applied to the power supply pins VDD of the system controller 129. The 5-volt supply 5V_EG is available from DC-to-DC converter 300 (FIG. 26a) by way of a ferrite bead inductor 381 (FIG. 12b). More particularly, a 5-volt supply signal 5V_CORE from the DC-to-DC converter is applied to the ferrite bead inductor 381, which, in turn, is used to generate the 5-volt supply signal 5V_EG. In order to stabilize the 5-volt supply signal 5V_EG, a plurality of bypass capacitors 1101-1111 (FIG. 13c) are connected between the 5-volt supply 5V_EG and system ground.

Please replace the paragraph beginning at page 34, line 5 through line 15 with the following amended paragraph:

A 3-volt power supply 3V_EG is also applied to the system controller 129 and, in particular, to the power supply pins VDD/3V. This 3-volt supply 3V_EG is also obtained from the DC-to-DC converter 300 (FIG. 26a) by way of a ferrite bead inductor 358. More particularly, 3-volt supply 3V_CORE, available at the DC-to-DC converter 300, is applied to the ferrite bead inductor 358, which, in turn, is used to generate the 3-volt power supply signal 3V-EG. A plurality of bypass capacitors 360, 362 and 364 are connected between the 3-volt supply 3V_EG and system ground for stabilizing.

Please replace the paragraph beginning at page 34, line 16 through line 28 with the following amended paragraph:

The system controller 129 is reset by a reset signal RCRST (FIG. 20) on power up. The reset signal RCRST is developed by the 3-volt power supply 3V_EG, available from the DC-to-

DC converter 300 (FIG. 26a) and circuitry which includes a resistor 359, a capacitor 361 and a diode 363. Initially on power up, the capacitor 361 begins charging up from the 3-volt supply 3V_EG through the resistor 359. During this state, the diode 363 is non-conducting. As the capacitor charges, the level of the reset signal RCRST rises to reset the system controller 129. Should the system be turned off or the 3-volt supply 3V_EG be lost, the diode 363 provides a discharge path for the capacitor 361.

Please replace the paragraph beginning at page 35, line 11 through line 26 with the following amended paragraph:

As mentioned above, the system controller 129 is capable of running at different clock frequencies, depending upon the voltage applied, while supplying a clock signal to the CPU 112. Even though the system controller 129 can supply either a 1X or a 2X clock signal to the CPU 112, the system controller 129 requires a 2X clock for proper operation. Thus, a 2X clock signal CLK2IN, available from a clock generator circuit 398 (FIG. 13d), is applied to the clock 2X pin CLK2IN of the system controller 129. In addition, 32 kilohertz (KHz) and 14 megahertz (MHz) clock signals are also applied to the system controller 129, available from the clock generator circuit 398, for proper operation. The system controller 129, in turn, provides a CPU clock signal CPUCLK to the CPU 112 and in particular to its clock 2-pin CLK2 by way of a resistor 1141 and the capacitors 1143 and 1145.

Please replace the paragraph beginning at page 35, line 27 through line 35 with the following amended paragraph:

The system controller 129 is adapted to be configured during an RC-RESET mode. In particular, the DRAM memory address lines MA[0..10], normally used for addressing the DRAM 111A (~~FIGS. 18 and 24~~ FIG. 18a), are pulled high or low in order to configure the system controller 129. More particularly, the DRAM memory address lines MA[0..10] are applied to either pull-up or pull-down resistors for configuration as illustrated in FIG. 17. Table 2 below illustrates the configuration shown.

Please replace the paragraph beginning at page 37, line 10 through line 19 with the following amended paragraph:

In order to conserve power, 3-volt DRAM 111A is used. The 3-volt power supply 3V RAM is applied to the VCC terminals of each of the DRAMS 111A. The 3-volt power supply 3V_RAM is available from the DC-to-DC converter 300 (FIG. 26a) by way of a ferrite bead inductor 440 (FIG. 18c). In particular, a 3-volt supply 3V_CORE available at the DC-to-DC converter 300 is applied to the ferrite bead inductor 440 to generate the 3-volt DRAM supply 3V RAM. A plurality of bypass capacitors 425-439 (FIG. 18c) are connected between the DRAM supply voltage 3V_RAM and system ground.

Please replace the paragraph beginning at page 37, line 20 through line 33 with the following amended paragraph:

The system controller 129 generates the appropriate row address strobes (RAS) and column address strobes for the DRAM 111A. In particular, the column address strobe lines CAS0[0..3] are applied to the upper and lower column address strobe pins (UCAS and LCAS) on the DRAM 111A by way of a plurality of coupling resistors 442 to 450 (FIG. 12f). Similarly, the row address signals RAS0 and RAS1 are applied to the row address strobe pins on the DRAM 111A by way of a plurality of coupling resistors 448 and 450. Writing to the DRAMS 111A is under the control of a DRAM write enable signal BRAMW, applied to the write enable pin WE on the DRAM 111A. The DRAM write enable signal BRAMW is generated by the system controller 129 by way of a coupling resistor 452.

Please replace the paragraph beginning at page 37, line 34 through page 38, line 9 with the following amended paragraph:

An EEPROM or NVRAM 111B (FIG. 12h) may be used to maintain system configuration parameters when the system is powered off. All user changeable parameters are stored in the EEPROM 111B. For example, pen calibration data and passwords, used during boot up, may be used in the EEPROM 452. The contents of the EEPROM 111B may be

shadowed into a CMOS memory when the system is active. Communication with the EEPROM 111B is under the control of the system controller 129 and in particular, a pair of programmable input/output pins GPI01 and GPI02. The GPI01 provides a clock signal to the EEPROM 111B while the pin GPI02 is used for data transfer.

Please replace the paragraph beginning at page 38, line 10 through line 16 with the following amended paragraph:

As discussed above, the wireless interface device 100 also includes the flash memory 117 (FIG. 25c), which is used for storing the BIOS. The system controller 129 allows for direct shadowing of the BIOS by enabling the appropriate address space to read the FLASH/DRAM write mode which allows all reads to come from the flash device with writes to the DRAM 111A memory devices.

Please replace the paragraph beginning at page 43, line 1 through line 11 with the following amended paragraph:

The integrated peripheral controller (IPC) 128 (FIG. [[14]]13a) is connected to the system data bus SD[0..15]. Addressing of the IPC 128 is accomplished by two bits SA0 and SA1 from the system address bus SA[0..23] and eight bits A[2..9] from the local address bus A[0..31]. The address bits from the local address bus A[2..8] are converted to 5 volts by way of a 3- to 5-volt signal converter 453 (FIG. 14e) to develop the 5-volt address signals XA[2..8]. A 32-kilohertz clock signal 32-KHz from the clock generator 398 (FIG. 13d) is applied to the clock input OSC1 of the IPC 128.

Please replace the paragraph beginning at page 43, line 12 through line 33 with the following amended paragraph:

Referring to ~~FIG. 20~~ FIGS. 20a-20e, in order to prevent spurious operation of the IPC 128 before the system power supply is stabilized, a power good signal PWRGOOD is applied to a power good pin PWRGD. The power good signal PWRGOOD is a delayed signal which assures that the 5-volt power supply has stabilized before the IPC 128 is activated. In particular,

a 5-volt power supply 5V_CORE is applied to a delay circuit which includes a resistor 454, a diode 456 and a capacitor 458. Initially, the 5-volt power supply signal 5V_CORE is dropped across the resistor 454. While the capacitor 458 is charging, the diode 456 is in a non-conducting state. As the capacitor 458 begins to charge, the voltage at the anode of the diode 456 increases as a function of the RC time constant. When the capacitor 458 is fully charged, it approaches the value of the power supply voltage 5V_CORE. When the capacitor 458 becomes fully charged, the power good signal PWRGOOD is applied to a power good pin PWRGD at the IPC 128 for enabling the IPC 128 after the power supply has stabilized. The diode 456 provides a discharge path for the capacitor 458 when the power supply is shut off. The power good signal PWRGOOD is also used to reset the keyboard controller 125.

Please replace the paragraph beginning at page 43, line 34 through page 44, line 13 with the following amended paragraph:

A 5-volt power supply 5V_CORE from the DC-to-DC converter 300 (FIG. 26a) is applied to a ferrite bead inductor 460 (FIG. 13c) to develop a 5-volt power supply 5V_206, which, in turn, is applied to the power supply pins VCC of the IPC 128. In order to delay application of the 5-volt power supply 5V_206 as discussed below, a charging circuit which includes a serially coupled resistor 462 and a capacitor 464 are connected between the power supply voltage 5V 206 and the system ground. A power supply reset signal PSRSTB, an active low signal, is applied to the junction between the resistor 462 and the capacitor 464 to discharge the capacitor 464 when the power supply is reset. Moreover, in order to stabilize the voltage of the power supply 5V_206, a plurality of bypass capacitors 466 and 468 are connected between the power supply 5V_206 and system ground.

Please replace the paragraph beginning at page 45, line 4 through line 17 with the following amended paragraph:

Interrupts by the system controller 129 and IPC 128 INTR_EG and INTR206 are applied to the CPU 112 by way of a diode 479 and pull-up resistor 481 (FIG. 14f). In particular, the

interrupt signals INTR_EG and INTR206 from the system controller 129 and IPC 128, respectively, are applied to the cathode of the diode 479 while the anode is pulled up to the power supply voltage 3V_CORE by the pull-up resistor 481. The logic level of the anode is set by the interrupt signal INTR, which is applied to the CPU 112. When the interrupt signals INTR206 and INTR_EG are high, the diode 479 does not conduct and the CPU 112 interrupt signal INTR will be high. When either of the interrupt signals INTR_EG or INTR206 are low, the diode 479 conducts, forcing the CPU 112 interrupt signal INTR low.

Please replace the paragraph beginning at page 46, line 25 through page 47, line 6 with the following amended paragraph:

The various clock signals used for the system are provided by the clock generator circuit 398 (FIG. 13d). The clock circuit 398 includes a clock generator, for example, an Integrated Circuit Designs Model No. ICD2028. A 14.318 MHz crystal 484 and a 32.768 KHz crystal 486 are applied to the clock generator 488. In particular, the crystal 484 is applied to a pair of X1 and X2 input pins along with a plurality of capacitors 489, 490, 492 and an input resistor 494. Similarly, the crystal 486 is applied to input pins XSYSB1 and XSYSB2. A pair of capacitors 496 and 498 are connected across the crystal 486.

Please replace the paragraph beginning at page 47, line 22 through line 27 with the following amended paragraph:

Selection of the various clock output signals is available by way of the select pins S0, S1 and S2. These pins S0, S1 and S2 are pulled up to the 3-volt power supply 3V_CORE by way of pull-up resistors 521, 523 and 525. The 3-volt power supply signal 3V_CORE is available from the DC-DC converter 300 (FIG. 26a).

Please replace the paragraph beginning at page 47, line 28 through page 48, line 6 with the following amended paragraph:

The clock generator 488 utilizes a 3-volt power supply CLOCK_VCC (FIG. 13d). The 3-volt power supply CLOCK_VCC is available from the DC-to-DC converter 300 (FIG. 26a) by

way of an in-line ferrite bead inductor 530. In particular, the 3-volt power supply 3V_CORE is applied to the ferrite bead inductor 530 to generate the power supply for the CLOCK_VCC for the clock generator 488. This power supply CLOCK_VCC is applied to the power supply pin VDD. The power supply signal CLOCK_VCC is also used as analog supply AVDD to the clock generator IC 488 and is applied to the analog supply AVDD by way of the resistor 532 and a pair of capacitors 534 and 536. The power supply signal CLOCK_VCC is also applied to the battery pin VBATT of the clock generator IC 488 by way of a diode 537 to prevent any back feeding.

Please replace the paragraph beginning at page 48, line 7 through line 24 with the following amended paragraph:

A number of the circuits in the system operate at either 3.3 volts or 5 volts. Thus, a plurality of bi-directional signal level translators 542 and 544 (~~FIG. 14~~FIGS. 14a and 14b) are provided, as well as the translator 453 previously discussed. The signal level translators 453, 542 and 544 may be as supplied by Integrated Circuit Technology, Model No. FCT164245T. Each of the signal level translators 453, 542 and 544 includes a 3-volt supply 3V_CORE and a 5-volt supply 5V_CORE, available from the DC-to-DC converter 300 (FIG. 26a). In order to stabilize the voltage of the 3- and 5-volt power supplies, 3V_CORE and 5V_CORE, a plurality of bypass capacitors are utilized. In particular, the bypass capacitors 546 through 552 are connected between the 3-volt supply 3V_CORE and system ground. Similarly, the bypass capacitors 554 through 560 are connected between the 5-volt supply 5V_CORE and system ground. The ground terminals of each of the signal level translators 542, 544 and 453 are also tied to system ground.

Please replace the paragraph beginning at page 50, line 16 through page 51, line 2 with the following amended paragraph:

Referring to ~~FIG. 15~~ FIGS. 15a-15g, the system includes a keyboard controller 125, which performs several functions, including battery monitoring, LCD status control, brightness and contrast control, as well as keyboard control. In addition, the system also maintains the

status of the remaining battery life, and also provides information to the system controller 129 when the battery voltage is low or other critical battery condition has occurred. In operation, the keyboard controller 125 will maintain the current status of the battery level until data is requested. When a critical battery condition event occurs, the keyboard controller 125 generates an SMI interrupt. As discussed above, the intelligent battery pack (IBP) 130 provides an indication of the percentage of remaining battery capacity. Communication between the IBP 130 and the keyboard controller 125 is by way of a bi-directional serial data bus, which includes a clock line BATCLK and a data line BATDATA. The data line BATDATA is a bi-directional line, which allows for bi-directional communication with the IBP 130. The clock line BATCLK is driven by the IBP 130, but may be pulled low by the keyboard controller 125.

Please replace the paragraph beginning at page 51, line 25 through line 35 with the following amended paragraph:

In addition to battery management, the keyboard controller 125 also supports an external PS/2-type keyboard, as well as a PS/2-type bar code reader, connected to a keyboard connector 140 (FIG. 29a). Communication between the keyboard or bar code reader (not shown) is by way of a standard type PS-2 two-wire bus connected to serial ports P4.6 and P4.7. In particular, the keyboard data KDATA is pulled up to the 5-volt voltage supply 5V_CORE by way of a pull-up resistor 582 while the keyboard clock signal KCLK is pulled up the 5-volt supply 5V_CORE by way of a pull-up resistor 584.

Please replace the paragraph beginning at page 52, line 1 through line 13 with the following amended paragraph:

Referring to ~~FIG. 29~~FIGS. 29a-29g, the keyboard connector 140 may be a 6-pin MINI-DIN connector or a DB-8 connector as shown. Pins 6-9 are connected to system ground. Pin 4 of the connector 140 is pulled up to the power supply voltage 5V_CORE by way of a fuse 579 and is filtered by a capacitor 581 and an inductor 583. The data signal KDATA is applied to pin 1 by way of a current-limiting resistor 585, while the clock signal KCLK is applied to pin 5 by

way of a current-limiting resistor 587 and a pair of capacitors 589 and 591. These clock and data signals KCLK and KDATA are connected to the ports P4.6 and P4.7, respectively, for serial communication with an external keyboard or bar code reader.

Please replace the paragraph beginning at page 52, line 14 through line 29 with the following amended paragraph:

Additionally, the keyboard controller 125 may be used to control the brightness level as well as the contrast level of the LCD display. More particularly, referring to ~~FIG. 27~~FIGS. 27a-27c, a contrast signal CONTRAST, available at port 0, pin 1 of the keyboard controller 125 (FIG. 15a) is used to adjust the contrast level of the LCD display. The contrast signal CONTRAST is applied to an adjustment terminal ADJ of a negative 24-volt DC voltage supply, which can be incrementally adjusted in steps by a 24-volt DC supply 586 (FIG. 27a), for example, a Maxim Model No. 749, which provides for 64-step adjustment. Thus, each high pulse will increment the contrast of the LCD display by one step. With a 64-step device, sixty-three pulses rolls the counter over and decreases the contrast by 1. The 24-volt DC supply 586 is under the control of an enable signal ENAVEE, available from the video controller 113A (FIG. 19f).

Please replace the paragraph beginning at page 52, line 30 through page 53, line 2 with the following amended paragraph:

In order to assure proper operation, the 24-volt supply 586 is connected in a circuit as shown in FIG. 27a, which includes a plurality of capacitors 588, 590, 592, 594; a plurality of resistors 596, 598, 600 an inductor 602; a PNP transistor 604; and a zener diode 606. The output of the circuitry is a nominal negative 24-volt signal LCDVEE, which is adjustable in 64 increments by way of the CONTRAST signal, as discussed above, to vary the contrast level of the LCD display.

Please replace the paragraph beginning at page 53, line 3 through line 15 with the following amended paragraph:

The keyboard controller 125 also controls the brightness of the LCD display. In particular, brightness adjustment signals BRIGHTNESS_UP, BRIGHTNESS_DOWN (FIG. 15a) are available at port 1, pins 6 and 7. These signals BRIGHTNESS_UP and BRIGHTNESS_DOWN are normally pulled up to the 5-volt supply 5V_KBD by way of a pair of pull-up resistors 608 and 610. These signals BRIGHTNESS_UP and BRIGHTNESS_DOWN are applied to a digital output potentiometer 612 (FIG. 27c), for example a Dallas Semiconductor Model No. DS1669-5O. The digital output potentiometer 612 is powered by a 5-volt power supply 5V_CORE, which is also used to pull up an unused output terminal, RH.

Please replace the paragraph beginning at page 53, line 23 through page 54, line 2 with the following amended paragraph:

The brightness control signal BRIGHTNESS from the digital output potentiometer 612, as well as a backlight control signal BACKLITEON and a backlight power signal BACKLITEPOWER are connected to the system by way of a 6-pin connector 615 (FIG. 27b). The backlight control signal BACKLITEON is connected to pin 4 of the connector 615 and pulled low by way of a pull-down resistor 617. The power control signal BACKLITEPOWER is applied to pins 1 and 2 while the backlight brightness control signal BRIGHTNESS is applied to pin 3. The backlight control signal BACKLITEON is available from the video controller 113A (FIG. 19f) and is used to power the backlight on the LCD. The backlight power signal BACKLITEPOWER, available from an FET 619 (FIG. 20b), is under the control of the backlight power control signal BACKLITEON, available from the video controller 113A (FIG. 19f).

Please replace the paragraph beginning at page 54, line 3 through line 34 with the following amended paragraph:

The FET 619 (FIG. 20b) is used to control power to both the LCD as well as the backlight. In particular, referring to FIG. 20b, the backlight power control BACKLITEON, is used to control an NPN transistor 617 by way of a current-limiting resistor 621. The NPN

transistor 621, in turn, is used to control the FET 619 to generate the backlight power signal BACKLITEPOWER at the drain terminal D1. The main power signal POWER (FIG. 28a) is connected to the collector of the NPN transistor 617 by way of a resistor 623. The main power signal POWER is also applied to a source terminal 51 of the FET 615. A gate terminal G1 of the FET 615 is connected between the resistor 623 and the collector of the NPN transistor 625. The backlight power control signal BACKLITEON is used to conserve power under certain power management conditions discussed above. This signal BACKLITEON controls the NPN transistor 625. In particular, in a normal state, the backlight power control signal BACKLITEON is high, which turns ON the NPN transistor 625. When the NPN transistor 625 is ON, the gate terminal G1 of the FET 619 is connected to system ground, which turns the FET 619 ON, thereby connecting the main power signal POWER to the drain terminal D1 of the FET 619 to provide a power signal BACKLITEIN, which is filtered by a ferrite bead inductor 625 (FIG. 28b) to provide the backlight power signal BACKLITEPOWER, that is applied to the LCD by way of the connector 615 (FIG. 27b). When the backlight power control signal is low, for example, during a power management mode, the NPN transistor 625 turns OFF, thereby connecting the gate G1 of the FET 619 to the main power signal POWER by way of the resistor 623, thereby turning the FET 619 OFF, disconnecting power to the LCD.

Please replace the paragraph beginning at page 54, line 35 through page 55, line 18 with the following amended paragraph:

The FET 619 may be supplied as a dual element with two FETs in a single package. As shown in FIG. 20b, the gate G2, source S2 and drain D2 terminals of the FET 619 are used to control power to the LCD, under the control of an LCD enable signal ENAVDD, available from the video controller 113A (FIG. 19f). In particular, the LCD enable signal ENAVDD is normally high and is de-asserted to disable the LCD power supply LCD_POWER. This LCD enable signal ENAVDD is pulled low by a pull-down resistor 627 and applied to an inverter 629, whose output is connected to the gate terminal G2 of the FET 619. The LCD power supply

signal LCD_VCC ~~(FIG. 19) is~~(FIGS. 19g and 19i) are applied to the source terminal S2 of the FET 619, while the drain terminal D2 represents the LCD power signal LCD_POWER, filtered by an inductor 629 and a capacitor 631. The LCD power signal LCD_POWER is connected to the LCD by way of the connectors 732 or 734 (FIG. 22b). In operation, the LCD power enable signal ENAVDD is high, which turns on the FET 619 to enable the LCD power supply LCD_POWER. When the LCD power enable signal ENAVDD is de-asserted, the FET 619 is turned OFF.

Please replace the paragraph beginning at page 55, line 19 through line 24 with the following amended paragraph:

The keyboard controller 125 (FIG. 15a) is connected to the system data bus SD[0..7]. The system address bit SA2 is used for addressing the keyboard controller 125. In particular, the address terminal of the keyboard controller 125 is connected to bit SA2 of the system address bus SA[0..23].

Please replace the paragraph beginning at page 55, line 25 through line 35 with the following amended paragraph:

Power to the keyboard controller 125 is provided by way of a 5-volt supply 5V_KBD, supplied to the power supply terminal VCC. The 5-volt supply 5V_KBD, provided by the DC-to-DC converter 300 (FIG. 26a) by way of an in-line ferrite bead inductor 618. In addition to supplying power to the keyboard controller 125, the 5-volt supply 5V_KBD is used to pull-up various pins by way of pull-up resistors 620, 622, 624, 626, 628, 630, 632 and 634. In order to stabilize the 5-volt power supply 5V_KBD, a plurality of bypass capacitors 636 and 638 are connected between the power supply 5V_KBD and system ground.

Please replace the paragraph beginning at page 56, line 1 through line 19 with the following amended paragraph:

As mentioned above, the keyboard controller 125 has various functions. One of those functions is to monitor when AC power is plugged into the machine from an AC adapter plug

633 (FIG. 29b), connected to the external power supply signal AC/DCIN by way of a pair of EM1 filters 641 and 643, and a connector 645. In particular, an AC power signal ACPWR, available from an FET 635 (FIG. 20e), is applied to port 3, pin 1 (FIG. 15a) by way of an inverter 636. The external power supply signal AC/DCIN, available from the AC plug 633, is used to control the gate terminal of the FET 635, normally pulled down a pull-down resistor 637. A 5-volt supply 5V_CORE is connected to the drain terminal while the source terminal is used for the AC power signal ACPWR, pulled down by a pull-down resistor 639. When an external power source is not connected to the FET 635, the signal ACPWR will be low. Once external power is connected to the connector 633, the signal AC/DCIN from the IBP 130 goes low, which, in turn, turns on the FET 635 to cause the signal ACPWR to go high.

Please replace the paragraph beginning at page 56, line 34 through page 57, line 5 with the following amended paragraph:

The video controller 113A (FIG. 19f) controls the video functions. The video controller 113A, for example, a model number CL-GD 6205 from Cirrus Logic, can support various video modes including a mono STN and a color TFT panel with up to 640 x 480 with 64 shades of gray. In addition, the video controller 113A will support 1024 by 768 resolution with 16 colors on a CRT through the aid of its on-board digital to analog converter.

Please replace the paragraph beginning at page 57, line 6 through line 26 with the following amended paragraph:

The video controller 113A utilizes two clock sources for timing, generated by an internal clock generator to produce the required frequencies for the display and memory timing. Two separate analog power supply sources AVCCMCLK and AVCCVCLK are provided to the analog power supply inputs AVCC1VCLK and AVCC4MCOK on the video controller 113A. These analog power supply sources AVCCMCLK and AVCCVCLK are derived from the 3-volt power supply 3V_CORE, available at the DC-to-DC converter 300 (FIG. 26a). In particular, the 3-volt power supply 3V_CORE is used to develop a 3-volt power supply VGA_VCC by way of

an in-line ferrite bead inductor 642. The power supply VGA_VCC, in turn, is filtered by a plurality of bypass capacitors 644-642, connected between the power supply VGA_VCC and system ground. The 3-volt power supply VGA_VCC is used to develop the analog power supplies AVCCMCLK and AVCCVCLK by way of a plurality of resistors 654 and 656 as well as a plurality of by pass capacitors 658 to 664, connected to an analog ground AGND. The analog ground AGND is tied to the digital ground GND by way of a ferrite bead conductor 664.

Please replace the paragraph beginning at page 57, line 27 through page 58, line 31 with the following amended paragraph:

The keyboard controller 125 also provides various miscellaneous system functions by way of its I/O ports 0, 1, and 3. Five port bits P0.0-P0.5 of port 0 are used for system control. Bit 0 is used to generate a signal KBC-P00, an active high signal, which disables the general purpose chip select signals GPCS1 and GPCS2, available at the system controller 129 (FIG. 12g) during boot-up, until the signals GPCS1 and GPCS2 are properly configured. As discussed above, the general purpose chip select signals GPCS1 and GPCS2 are used for selecting the pen controller 110A (FIG. 21e), the radio interface 114B (FIG. 16b) and the UART (134). Bit P0.1 is used to generate a contrast signal CONTRAST, normally pulled low down by a pull-down resistor 639 (FIG. [[5]]15e) for contrast control of the LCD as discussed above. Briefly, the contrast signal CONTRAST is used to step the 24-volt supply 586 (FIG. 27a). Bit P0.2 is used to generate a keyboard shutdown signal KBSHUTDOWN. This signal KBSHUTDOWN, discussed below, is active low, and in conjunction a pen shutdown signal PEN_SHUTDOWN, available at the pen controller 110A (FIG. 21e), is used to generate a shutdown signal SHUTDOWN to shutdown the AC-to-DC converter 300 (FIG. 26a) during low power conditions. More particularly, the keyboard shutdown signal KBSHUTDOWN, pulled up by a pull-up resistor 641, and the pen shutdown signal PEN_SHUTDOWN, pulled low by a pull-down resistor 643, are diode ORed by a pair of diodes 645 and 647. The cathodes of the diodes 645 and 647 are joined to form the active low shutdown signal SHUTDOWN. If the keyboard

shutdown signal KBSHUTDOWN is asserted, the shutdown signal SHUTDOWN will be forced low, which, in turn, is used to disable the DC-to-DC converter 300 (FIG. 26a). Bit P0.3 is used to generate a signal FLASHVPP to enable the flash memory devices 742-748 (~~FIG. 25~~FIGS. 25a-25c) to be programmed. In particular, when the signal FLASHVPP is low, the flash memory devices 742-748 can be programmed. Bit P0.4 is used to generate a signal KBC_P04. The signal KBC_P04 is an active high signal and is used to indicate to the system controller 129 (FIG. 12g) that a low battery condition has occurred. Bit P0.5 is used for speaker control as discussed above. The pen P0.5 is used to generate the speaker disable signal SPKRDISABLE, an active high signal.

Please replace the paragraph beginning at page 58, line 32 through page 59, line 8 with the following amended paragraph:

Port 1, bits P1.1, P1.5, P1.6, and P1.7 of the keyboard controller 125 are used for system functions. Bit P1.1 is configured as an input and is used to indicate to the keyboard controller 125 that the system is in a test mode. As discussed above, the test mode signal TEST_MODE is used to enable the flash memory device 742 (FIG. 25a) to be programmed. In particular, as discussed above, the test mode signal TEST_MODE is used to generate a decode signal FLIP_SA18 (FIG. 17d) for decoding of the flash memory device 742. Port 1, bits P1.5, P1.6, and P1.7 are used for LCD control. In particular, the pen P1.5 may be used for LCD status control. the pens P1.6 and P1.7 are used for brightness control of the LCD as discussed above.

Please replace the paragraph beginning at page 59, line 9 through line 29 with the following amended paragraph:

Port 3, bits P3.1, P3.2, P3.3, P3.4, P3.5, and P3.7 are configured as inputs. As discussed above, a signal ACPWR, available from the source of the FET 635 (FIG. 20e), is applied to the pin P3.1. This signal ACPWR notifies the keyboard controller 125 that an external power source is connected to the system. The signal CD_LED is applied to the pin P3.2. This signal, CD_LED, available from the radio interface (FIG. 16a), indicates that the radio is receiving a

signal. A signal TX/RX_LED, also available from the radio interface, is applied to the pin P3.3. This signal TX/RX_LED indicates that the radio is in a transmit mode. A signal DOCKACK/: may be applied to the pin P3.4. This signal may be used to indicate to the keyboard controller 125 that a device is docked to the UART 134. The development of the signal DOCKACK/: does not form a part of the present invention. A second test mode signal TEST MODE_2 may be applied to the pin P3.5 for added functions. A signal PC5_P37 is applied to the pen P3.7. This signal PC5_P37 is available from the system controller 129 (FIG. 12g) and indicates that the system is in a sleep state as discussed above.

Please replace the paragraph beginning at page 60, line 35 through page 61, line 11 with the following amended paragraph:

As will be discussed in more detail below, the system also includes an LCD controller to control the LCD screen 113C. The power supply for the LCD controller LCD_VCC can likewise be supplied as either three volt or five volt by way of the 3- and 5-volt power supply voltages 3V_CORE and 5V_CORE, available at the DC-to-DC converter 320 (FIG. 26a). Depending on the voltage selected, only one of the component locations 684 and 686 will be populated to provide the LCD power supply voltage LCD_VCC. In addition, a power supply voltage VGABUS_VCC is used for the VGA bus. This power supply voltage VGABUS_VCC is generated by the DC-to-DC converter 320 by way of a ferrite bead inductor 688.

Please replace the paragraph beginning at page 61, line 29 through page 62, line 6 with the following amended paragraph:

The video controller 113A requires two separate clock signals: 14 MHz; and 32 KHz. The 14 MHz clock signal is used for most timing including the LCD panel memory and the bus cycle while the 32 KHz clock signal is used for video memory refreshing when the system is suspended. These clock signals are supplied by the clock generator 398 (FIG. 13a) by way of the signal level translator 452 (FIG. 14e). More particularly, 32 KHz and 14 MHz clock signals 32 KHz and 14 MHz from the clock generator 398, respectively, are applied to the signal level

translator 452 to transform these respective signals into 5-volt signals 32 KHz_5V an 14 MHz_5V to provide a suitable clock signal voltage for the video controller 113A.

Please replace the paragraph beginning at page 62, line 7 through line 14 with the following amended paragraph:

RGB data from the video controller 113A (FIG. 19f) is supplied to the LCD screen 113C by way of a data bus PDATA[0..17]. This data bus PDATA[0..17] is applied to a plurality of current limiting resistors 708-742, respectively, to generate the buffer signals PDBUF[0..17]. These buffer signals PDBUF[0..17] are connected to the LCD panel 113 along with various control signals by way of a pair of connectors 732 and 734.

Please replace the paragraph beginning at page 62, line 15 through line 36 with the following amended paragraph:

The BIOS as well as other data is stored in flash memory, for example, 512K by 8-bit memory devices 742-748 (~~FIG. 25~~FIGS. 25a-25c). These flash memory devices 742-748 are connected to the local ISA bus 150 by way of the system address bus SA[0..23] and the system data bus SD[0..15]. The chip enable pins CE of the flash memory devices 742-748 are selected by a decoder circuit (~~FIG. 17~~FIGS. 25a-25c), as will be discussed in more detail below. The output enable pins OE on the flash memory devices 742-748 are under the control of a memory read signal MEMR. The memory read signal MEMR is under the control of the system controller 129. The write/enable pins WE, which are active low, are under the control of a memory right gate signal MEMWGATE. This signal MEMWGATE is only enabled when the flash memory devices 742-748 are being programmed. As discussed above, programming of the flash memory devices 742-748 is under the control of a flash program signal FLASHVPP, available at port 0.3 of the keyboard controller 125 (FIG. 15a). This programming signal FLASHVPP, normally pulled high by a pull-up resistor 749 (FIG. 17c), is ORed with a memory write signal MEMW by way of an OR gate 751 to generate a signal MEMGATE, an active low signal.

Please replace the paragraph beginning at page 63, line 1 through line 7 with the following amended paragraph:

The power supply for the flash memory devices 742-748 is developed by a 5-volt power supply signal 5V_ROM. The 5-volt power supply signal 5V_ROM is available from the DC converter 300 (FIG. [[20]]26a) by way of a ferrite bead inductor 751. This power supply signal 5V_ROM is also connected to a plurality of by-pass capacitors 752-758, for stabilization.

Please replace the paragraph beginning at page 63, line 8 through line 26 with the following amended paragraph:

Decoding of the flash memory devices 742-748 is provided by the circuitry that includes the buffers 760, 762, the inverters, 764, 766, and 768 and OR 770 and a 3- to 8-bit multiplexer, Model No. 74HCT138, for example, as manufactured by Motorola and a pair of resistors 772 and 774 (FIG. 17d). In particular, the system address bits SA[19..21] are applied to a 3- to 8-bit multiplexer 776. The system address bit SA18 is applied to the inverter 760 to develop a FLIP_SA18 signal that is pulled down by the pull-down resistor 774. During a normal boot-up, the FLIP_SA18 signal will be same as the system address bit SA18. However, during a test mode boot-up, the FLIP_SA18 signal will be low until a control signal available at the control signal GPI00, available at the system controller 129, goes low in order to enable the system to boot from the BIOS in the flash memory device 742 as will be discussed in more detail below. Once the GPI00 signal goes low, the FLIP_SA18 signal will be the same as the system address bit SA18.

Please replace the paragraph beginning at page 64, line 13 through line 23 with the following amended paragraph:

In order to provide the ability of the system to update the BIOS in the flash memory device 742 and to recover from a corruption of the BIOS data in the flash memory device 742, a uniform asynchronous receiver transmitter (UART) 788 (FIG. 23b) is provided. The UART 788 is connected to the system data bus SD[0..15] and the system address bus bits SA[0..2]. The

UART 788 is powered by the 5-volt power signal 5V_CORE, available at the DC-to-DC converter [[320]]300 (FIG. 26a). A 1.84 MHz clock signal, 1.84 MHz_5V, available at the signal level translator 452, is used to drive the UART 788.

Please replace the paragraph beginning at page 64, line 24 through page 65, line 2 with the following amended paragraph:

A serial interface 790 (FIG. 30a), consisting of a standard DB-9 connector, enables external serial data to be received by the UART 788 (FIG. 23b). The UART signals are filtered by way of a plurality of resistors 792-806 and bypass capacitors 802-822 and applied to an optional disaster recovery adapter 824, an RS-232 interface, connected to the rear of the DB-9 connector 790 and permits the flash memory devices 742-748 (~~FIG. 25~~FIGS. 25a-25c) to be updated by an external source in the event of a flash disaster. The flash recovery adapter 824 may be implemented as a DB-9 connector and is connected to the 5-volt power supply 5V_CORE, which, in turn, is connected to a plurality of bypass capacitors 826 and 828. An additional four capacitors 830-836 are connected to the module 824 as shown.

Please replace the paragraph beginning at page 65, line 3 through line 11 with the following amended paragraph:

The power supply for the system includes the DC-to-DC converter 300 which has the ability to provide both 3-volt and 5-volt power supplies signals to the various subsystems as discussed. The DC-to-DC converter includes a switching power supply 850, for example, a Maxim type 786. One source of power to the DC-to-DC converter 300 is the IBP 130, for example, 7.2 volts nominal, as well as from an external source of AC power connected to the plug 633 (FIG. 29b).

Please replace the paragraph beginning at page 65, line 12 through line 33 with the following amended paragraph:

Input power to the DC-to-DC converter 300 may be from an AC/DC converter (not shown) connected to the plug 633, which has a DC output voltage between 5.5-15 volts DC,

applied to a power supply terminal AC/DCIN (FIG. 28a) as well as internal batteries, for example, the IBP 130, connected to the system by way of a connector 850 (FIG. 26c). The battery supply voltage from the IBP 130 is connected to the battery positive terminal BATT (FIG. 28a). The two supplies BATT and AC/DCIN are alternatively used to develop a main power signal POWER (FIG. 28a), that is applied to a switching power supply 851, for example, a Maxim type 786 by way of a pair of FETS 854 and 856 ~~(IL. 26)~~(FIG. 26a), under the control of a main power switch 855 (FIG. 28a). The main power signal POWER is applied to a drain input D2 on each of the FETS 854 and 856. A bypass capacitor 860 is connected to the drain terminal D2 of the FET 856 and system ground. The source terminals S2 of each of the FETS 854 and 856 is connected to the switching power supply 851 to provide 5- and 3-volt references by way of the zener diodes 860 and 862, respectively. The gate terminals G1 and G2 of the FETS 854 and 856 are under the control of the switching power supply 851.

Please replace the paragraph beginning at page 66, line 8 through line 17 with the following amended paragraph:

The frequency of the switching power supply 851 is under the control of a pair resistors 888 and 890 and a capacitor 892, connected to the SYNC and reference terminals on the switching power supply 851. A HOOK-VCC signal is applied to the VH and VL pins of the switching power supply 851. This signal HOOK-VCC is available from the module 894 (FIG. 29g), discussed above. The signal HOOK-VCC signal is connected to the switching power supply 851 by way of a resistor 896 (FIG. 26a); a plurality of capacitors 898, 900 and 902; and an FET 904.

Please replace the paragraph beginning at page 66, line 18 through line 25 with the following amended paragraph:

As mentioned above, both the pen controller 110A (FIG. 21e) and keyboard controller 125 (FIG. 15a) are used to develop a shutdown signal SHUTDOWN. The shutdown signal SHUTDOWN is pulled low by a pull-down resistor 906 and applied to an active low shutdown

pen SHDN* on the switching power supply 851. The shutdown signal SHUTDOWN (FIG. 20a) is indicative of a shutdown by the keyboard controller 125 (FIG. 15a).

Please replace the paragraph beginning at page 66, line 26 through page 67, line 5 with the following amended paragraph:

As mentioned above, one source of power for the system is the IBP 130 which accounts for temperature and discharge rates and sends it to the keyboard controller 125 (FIG. 15a). Two predefined levels are set in the IBP 130 to indicate low battery and critical battery. The IBP 130 will inform the keyboard controller 125 of a low battery when there is approximately five minutes left. When the battery charge is between 5 minutes to 2 minutes, the IBP 130 will report a battery critical condition. Within the final thirty seconds the IBP 130 will force an immediate shutdown. The IBP 130 will report the battery status approximately once every 2.5 seconds. If the system is changing to a power savings mode, a command will be sent to the IBP 130 to put the IBP 130 into a power-saving state. The IBP 130 will tri-state its communication lines and discontinue reporting battery status to the system.

Please replace the paragraph beginning at page 67, line 6 through line 12 with the following amended paragraph:

A charge control signal CHGCTRL from the IBP 130 is used to control charging. Referring to FIG. 28a, the charge control signal CHGCTRL is applied to a zener diode 910, for example, a 5.1V zener diode. The zener diode 910 controls whether the IBP 130 is fast charged or trickle charged as a function of the magnitude of the charge control signal CHGCTRL.

Please replace the paragraph beginning at page 68, line 30 through page 69, line 7 with the following amended paragraph:

As mentioned above, the wireless interface device 100 includes a radio system which allows for wireless interfacing with a host computer and also wireless interfacing to both a wired local area network (LAN) and a wireless LAN. The radio subsystem has been discussed above. It is implemented by way of an interface 960 (FIG. 16a), implemented by way of a 25 x 2 header,

which connects the radio subsystem to the balance of the circuitry in the wireless interface device 100. In particular, the system data bus SD[0..15], as well as the system address bus bits SA[0..2] are connected to the interface 960. The radio interface 960 is under the control of the system controller 129 (FIG. 12g), such as I/O write (IOW), I/O read (IOR) and an address enable signal (AEN).

Please replace the paragraph beginning at page 69, line 8 through line 20 with the following amended paragraph:

Output signals from the radio interface 960 include the signals CD_LED, TX/RX_LED, IRQ10 and IOCS16. As discussed above, the signal CD_LED indicates a connection has been made with a host computer 101. The signal TX/RX_LED indicates that a signal is either being sent or received through the radio interface 960. As mentioned above, the peripheral controller 128 (FIG. 13<u>a</u>) is responsible for interrupt control. Thus, the radio subsystem interrupt IRQ10 is applied to the peripheral controller 128. Power supply for the radio interface 960 is by way of a 5-volt power supply signal 5V_CORE, available at the DC-to-DC converter 300 (FIG. 26<u>a</u>), which is filtered by a pair of bypass capacitors 962 and 964.

Please replace the paragraph beginning at page 69, line 21 through line 29 with the following amended paragraph:

The interrupts for both the radio interface 960 IRQ10, as well as the UART 788 (FIG. 23<u>b</u>) IRQ4, are formed into a common signal IRQ10/4 and applied to the system controller 129 by way of a resistor 966. In particular, the radio interface interrupt signal IRQ10 is applied to an inverter 962, whose output is ORed by way of the OR gate 964 with the UART 788 interrupt signal IRQ4. The output of the OR gate 964 forms the combined interrupt signal IRQ10/4.

Please replace the paragraph beginning at page 69, line 30 through page 70, line 14 with the following amended paragraph:

The radio interface 960, as well as the UART 788 (FIG. 23<u>b</u>), are selected by the chip select signals RADIOCS and URTCS. These signals are available at the output of a pair of the

OR gates 968 and 970, respectively. The system address bit SA3 is inverted by way of an inverter 972 and ORed with a general purpose chip select gate signal GPCS1GATE by way of the OR gate 970 to generate the UART chip select signal UARTCS. The system address bit SA3 is applied directly to the OR gate 968 and ORed with the general purpose chip select gate signal GPCS1GATE to generate the radio chip select signal RADIOCS. The general purpose chip select signal gate signal GPCS1GATE is available at the output of an OR gate 974. In particular, a general purpose chip select signal GPCS1, available from the system controller 129 (FIG. 12g), is ORed with an output from pin 0 of port 0 of the keyboard controller 125 (FIG. 15a) to cause the radio interface 960 to be addressed at addressed 3EO-3E7 and the UART 788 to be addressed at address 3EA-3EF. The signal KBC_P00 is normally pulled up to the 5-volt power supply voltage 5V_CORE by way of a pull-up resistor 976.

Please replace the paragraph beginning at page 70, line 15 through page 71, line 14 with the following amended paragraph:

The pen controller 110A is illustrated in FIG. 21b and is adapted to cooperate with an analog-resistive type digitizer 106. The pen controller 110A includes a controller 980, for example a Motorola type MC68HC705J2 microcontroller, with the firmware being programmed within the part. The controller 980 communicates with the system by way of the system data bus SD[0..15]. In particular, serial data from a port PB6 on the controller 980 is applied to a shift register 982, which, in turn, is connected to an 8-bit parallel buffer 984, which, in turn, is connected to the serial data bus SD[0..15]. The controller 980 is adapted to be used with an analog-resistive touch screen digitizer, for example a drawing No. 8313-34 Rev. C4, as manufactured by Dynapro. XY information from the digitizer 106 is received by the controller 980 by way of a connector 986. The X and Y information from the digitizer is connected to a 12-bit analog-to-digital (A/D) converter and also applied to port PA5 of the microcontroller 980. In particular, the X- data from the digitizer is applied to the A1 terminal of the A/D converter 988 by way of a pull-up resistor 990 and an FET 992. The FET 992 is under the control of a

charge pump 994, for example a Linear Technology Model No. LTC1157C58. The Y- data from the digitizer is applied to the terminal A1 of the A/D converter 988 by way of a current-limiting resistor 994. A pair of bypass capacitors 996 and 998 are tied between the terminals A0 and A1 of the A/D converter 988 and an analog ground PEN_AGND. The X+, Y+, X-, Y- inputs from the digitizer are also applied to the controller ports PA[0..4] by way of a plurality of transistors 1000, 1006, 1010, 1016 and 1018; a plurality of resistors 1002, 1008, 1012, 1014, 1020, 1022, 1028, 1032 and 1034; an inductor 1004; and a plurality of capacitors 1024 and 1026. The transistor 1018, as well as the transistors 992 and 998, are used to prevent leakage in a suspend state.

Please replace the paragraph beginning at page 71, line 15 through line 26 with the following amended paragraph:

Power from both analog and digital power supply and grounds are supplied to the system. In particular, a 5-volt digital power supply PEN_VCC, developed from the 5-volt supply 5V_CORE, is available from the DC-to-DC converter 300 (FIG. 26a) by way of an in-line ferrite bead inductor 1028. An analog power supply PEN_AVCC is developed from the digital supply PEN_VCC by way of an in-line ferrite bead inductor 1030. The digital power supply PEN_VCC is applied to the microcontroller 980 and filtered by a bypass capacitor 1030. The analog supply PEN_AVC is utilized by the 12-bit analog-to-digital converter 988 and filtered by way of a bypass capacitor 1032.

Please replace the paragraph beginning at page 71, line 33 through page 72, line 10 with the following amended paragraph:

A 5-volt signal PENACT_5V, available at the port P5V pin of the microcontroller is converted to a 3-volt signal PENACT_3V by way of a pair of voltage dividing resistors 1042 and 1044. This signal PENACT_3V is applied to a 3-volt terminal of the system controller 129 (FIG. 12g). As discussed above, the power supply for the FETs 992 and 1018 is provided by the charge pump 994. The power supply for the charge pump 994 is a 5-volt power supply signal

5V_CORE, available at the DC-to-DC converter 300 (FIG. 26a). A ground terminal of the charge pump 994 is connected to system ground by way of a pull-down resistor 1050. The 5-volt power supply PEN_VCC is also utilized by the shift register 982 and the data buffer 984 and buffered by way of a pair of bypass capacitors 985 and 987.

Please replace the paragraph beginning at page 72, line 11 through line 16 with the following amended paragraph:

The chip select signal PENCS for the data buffer 984 is generated by an OR gate 1052. The general purpose chip select signal GPCS2 is available from the system controller 129 (FIG. 12g), as well as a signal KBC_P00, available from the keyboard controller 125 (FIG. 15a) are applied to the inputs of the OR gate 1052.

Please replace the paragraph beginning at page 72, line 17 through line 31 with the following amended paragraph:

A pen shut-down signal PEN_SHUTDOWN is used to develop a shut-down signal SHUTDOWN as discussed above for turning on the switching power supply 851 (FIG. 26a). The pen shutdown signal PEN_SHUTDOWN is developed by the circuit that includes the transistors 1060, 1062 and 1064; a plurality of resistors 1066, 1068, 1069, 1070 and 1072; and a capacitor 1074. In particular, a 5-volt power supply signal 5V_CORE is applied to a pair of voltage-dividing resistors 1070 and 1072, which, in turn, is used to bias the transistor 1064 on. The base-emitter voltage is held fairly constant by the capacitor 1074. Once the transistor 1064 is turned on, it is used to control the FET 1062. A main power supply signal POWER is applied to the gate of the FET 1062 by way of the resistor 1069. Wake up of the system by way of the pen subsystem is discussed below.

Please replace the paragraph beginning at page 72, line 33 through page 73, line 4 with the following amended paragraph:

As mentioned above, the wireless interface device 100 includes the flash memory devices 742-748 (~~FIG. 25~~FIGs. 25a-25c). As will be discussed in more detail below, the flash memory

devices enable user software upgrades by way of the radio interface 960 (FIG. 16a). Should power be lost during the programming, the data within the flash memory devices 742-748 will be corrupted, which could result in the system failing to boot.

Please replace the paragraph beginning at page 73, line 5 through line 20 with the following amended paragraph:

In order to enable recovery from such a condition, recovery BIOS is stored in a protected sector of the flash memory device 742, which will be unaffected during reprogramming. In addition, a serial port interface 790 (FIG. 30a) is provided to enable the flash memory devices 742-748 to be programmed in such a condition by an alternative wired source following a normal boot-up. Unfortunately, the configuration of the flash memory device 742 may result in the system failing to boot. More particularly, disaster recovery BIOS is not stored at the uppermost address of the flash memory device 742. Each flash memory device 742-748 are 512K x 8-bit devices. With reference to Table 5 above, the flash memory device 742 is mapped to the address range $0C0000-$0FFFFF. The recovery BIOS is contained in the lower half of that range (i.e. $0E0000-$0FFFF).

Please replace the paragraph beginning at page 73, line 21 through line 36 with the following amended paragraph:

On a normal boot-up, the system begins executing code at the top of the address range (i.e. $0C0000-ODFFFF) flash memory device 742 by way of the system address bit SA18. More particularly, on a normal boot-up a test mode signal TEST_MODE, available at port 1.1 of the keyboard controller 125 (FIG. 15a) is pulled high by the keyboard controller 125 during boot-up, which enables the buffer 762 (FIG. 17d) which, in turn, enables another buffer 760 to enable the system address bit SA18 during boot-up. When the system address bit SA18 is enabled, the system begins executing code at the top of the address range ($0C0000) of the flash memory device 742. However, during a condition when the data in the top half of the address range

($0C00000-0DFFFFF) becomes corrupt as a result of a problem occurring during reprogramming, the system may not boot during such a condition.

Please replace the paragraph beginning at page 74, line 1 through line 12 with the following amended paragraph:

In order to solve this problem, the system address bit SA18 is forced low. By forcing the system address bit SA18 low, the system will begin executing code from the protected area of the flash device 742 in the address range ($0E0000-$0FFFF) during such a condition where the disaster recovery BIOS resides in a protected sector. In particular, the system address bit SA18 is applied to the buffer 760 (FIG. 17d), which is under the control of the test mode signal TEST_MODE by way of the buffer 762. The output of the buffer 760 is a signal FLIP_SA18, which is applied to the address pin A18 (FIG. 25a) on the flash memory device 742.

Please replace the paragraph beginning at page 74, line 13 through line 23 with the following amended paragraph:

During a normal boot-up, the test mode signal TEST_MODE will enable the buffer 762 (FIG. 17d) and, in turn, the buffer 760 to cause the system address bit SA18 to drive the signal FLIP_SA18. During a condition when the code in the flash memory device 742 becomes corrupt, the test mode signal TEST_MODE is forced low, which, in turn, forces the signal FLIP_SA18 low, resulting in the system executing code from the protected area (i.e. $0E0000-0FFFF) of the flash memory device 742 during such a condition to enable the flash memory device 742 (FIG. 25a) to be reprogrammed by way of the serial interface 790 (FIG. 30a).

Please replace the paragraph beginning at page 74, line 24 through page 75, line 5 with the following amended paragraph:

There are various ways in which to force the test mode signal TEST_MODE low during reprogramming of the flash memory device 742 by way of the serial interface 790. One way is to externally ground the test mode signal TEST_MODE during such a condition. In particular, the test mode signal TEST_MODE may be connected to one pin of a two-pin header 1100 (FIG.

30c). The other pin of the header 1100 is connected to system ground. During reprogramming of the flash memory device 742, an external jumper (not shown) is inserted into the header 1100 to shunt the test mode signal TEST_MODE to system ground to enable the system to execute code from the protected or boot block area of the flash memory device 742 in order to enable the system to be booted. Once the system is booted, the flash memory device 742 is reprogrammed by way of the serial interface 894 (FIG. 29g). Once reprogramming is complete, the shunt is removed from the header 1100 (FIG. 30c) and the adapter plug 790 is removed, restoring the system to normal operation.

Please replace the paragraph beginning at page 75, line 7 through line 14 with the following amended paragraph:

In order to conserve battery power, the wireless interface device 100 goes into a suspend mode when the system is not in use. As discussed above, a shut down signal SHUTDOWN (FIGS. 20a and 26a) is used to shut down the power supply 851 (FIG. 26a) during such a condition, which essentially disables the power to all but the circuitry required to detect a pen down event by way of the main power signal POWER (FIG. 28a).

Please replace the paragraph beginning at page 75, line 15 through line 26 with the following amended paragraph:

Three sources control the shut down signal SHUTDOWN: the keyboard controller 125 (FIG. 15a); the pen controller 110A (FIG. 21e) and a signal HOOK_VCC, connected to the switching power supply 851 (FIG. 26a) by way of the FET 904. These sources are diode ORed to the shut down signal SHUTDOWN by way of the diodes 645 and 647 (FIG. 20a) and a diode 1102 (FIG. 28c). During a normal state, the shut down signal SHUTDOWN is high, which enables the power supply 851 (FIG. 26a). When the shut down signal SHUTDOWN goes low, the power supply 851 goes into an inactive state. During the inactive state, minimum power is supplied to the pen detection circuitry as discussed above.

Please replace the paragraph beginning at page 75, line 27 through line 32 with the following amended paragraph:

As will be discussed in more detail below, once the system is turned on by the main power switch 855 (FIG. 28a), the shut down signal SHUTDOWN will be under the control of the pen shutdown signal PEN_SHUTDOWN, available from the pen controller 110A (FIG. 21e) and the keyboard controller shut down signal KBSHUTDOWN (FIG. 20a).

Please replace the paragraph beginning at page 75, line 33 through page 76, line 6 with the following amended paragraph:

The keyboard controller 125 (FIG. 15a) can place the system in a suspend state by way of a command, which, in turn, causes the keyboard controller shut down signal KBSHUTDOWN, available at port P0.2, to go low. More particularly, during normal operation, only the keyboard shutdown signal KBSHUTDOWN is high, placing control of the suspend state solely in the keyboard controller 125. The keyboard controller 125 can then force the system into a suspend state by forcing port P0.2 low, which, in turn, places the power supply 851 (FIG. 26a) in an inactive state.

Please replace the paragraph beginning at page 76, line 7 through page 77, line 8 with the following amended paragraph:

The pen shut down control signal PEN_SHUTDOWN is used to wake the system from a suspend state. More particularly, as mentioned above, during a suspend state, power from the main power supply POWER (FIG. 28a) is applied to the collector of the transistor 1064 (FIG. 21d) and to the drain of the FET 1062. Since the 5-volt power supply 5V_CORE is unavailable during a suspend state, the transistor 1064 will be OFF, allowing power to appear at the gate of the FET 1062, thus turning the FET 1062 ON. Once the FET 1062 is turned ON, the main power signal POWER is applied to the XPLUS terminal of the digitizer panel. Thus, a pen (or finger) down event will result in the YPLUS terminal being connected to the XPLUS terminal by way of a finite resistance (i.e. 500-1500 Ohms) to apply power to the YPLUS terminal, which, in turn, is

connected to the drain of the P-channel FET 1060 while its source is used as the pen shutdown signal PEN_SHUTDOWN. The FET 1060 is under the control of a leakage signal LEAKAGE, available at the output of the charge pump 994. Since the leakage signal LEAKAGE will be low during a suspend state, the FET 1060 will turn on in response to the pen down event, thereby connecting the YPLUS terminal to the pen shut down signal PEN_SHUTDOWN. As mentioned above, the YPLUS terminal will be high in response to a pen down event following a suspend state. As such, the pen shut down signal PEN_SHUTDOWN will go high. Since the pen shut down signal PEN_SHUTDOWN is diode ORed with the shut down signal SHUTDOWN, the shut down signal SHUTDOWN will thus be forced high in response to a pen down event following a suspend state, which, in turn, will wake up the power supply 851 (FIG. 26a). Once the system is wakened, the keyboard controller shutdown line KB_SHUTDOWN goes high, latching the system ON. The resistors 1070, 1072 and the capacitor 1074 are used to delay turning ON the transistor 1064 and the turning OFF of the FET 1062 before the keyboard shutdown signal KB_SHUTDOWN is pulled high which would cause the pen shut down signal PEN_SHUTDOWN to go low before the keyboard shutdown signal KB_SHUTDOWN goes high.

Please replace the paragraph beginning at page 77, line 23 through page 78, line 35 with the following amended paragraph:

The system ON/OFF switch 855 (FIG. 28a) enables the system to be completely shut off. When the switch 855 is closed, power from either the IBP 130 or the external AC-to-DC converter supplies power to the system. In order to wake up the system from an OFF state, a shutdown line SHUTDOWN must be held high until the keyboard controller 125 pulls its shutdown pin KB_SHUTDOWN high. As discussed above, the keyboard shutdown signal KB_SHUTDOWN is diode ORed relative to the shutdown signal SHUTDOWN, which controls the power supply 851 (FIG. 26a). Until the time when the keyboard shutdown signal KB_SHUTDOWN is pulled high, a signal HOOK_VCC is used to force the shut down signal

SHUTDOWN high. As mentioned above, the HOOK_VCC signal is also diode ORed relative to the shutdown signal SHUTDOWN by way of the diode 1102 (FIG. 28c). However, for proper operation of the system, the shutdown signal SHUTDOWN will be under the control of the keyboard controller 125 (FIG. 15a) after the system is turned on. Thus, a 5-volt power supply signal HOOK_VCC, available at the power supply 851 (FIG. 26a), forces the shut down signal SHUTDOWN high until the keyboard controller 125 (FIG. 15a) has time to pull its keyboard shutdown signal KB_SHUTDOWN high. The 5-volt power supply signal HOOK_VCC is always high when the main power switch 855 is turned on. On power-up, the 5-volt power supply signal HOOK_VCC forces the shutdown signal SHUTDOWN (FIG. 28c) high by way of an FET 1104 and the diode 1102, which, in turn, wakes up the power supply 851 (FIG. 26a). Once the power supply 851 is enabled, a power supply signal MAX 786_VCC is used to turn off the FET 1104 to place the control of the shut down signal SHUTDOWN under the control of the keyboard controller 125 as discussed above. In order to provide sufficient time for the keyboard controller 125 to pull its keyboard shutdown signal KB_SHUTDOWN high, the turn OFF of the FET 1104 is delayed by way of a resistor 1106 and a capacitor 1108. In particular, once the main power switch 855 is closed, the power supply signal MAX786_VCC will be low, thereby causing the FET 1104 to be turned ON, which connects the power supply signal HOOK_VCC to the shutdown signal SHUTDOWN by way of the diode 1107. Once the power supply 851 is enabled, the signal MAX786_VCC, applied to the gate of the FET 1104, turns off the FET 1104, placing the shutdown signal SHUTDOWN under the control of the keyboard controller shutdown signal KB_SHUTDOWN as discussed above. The resistor 1106 and capacitor 1108 delay the turning off of the FET 1104 after the signal MAX786_VCC goes high for a sufficient time to allow the keyboard controller 125 to pull its keyboard shut down signal KB_SHUTDOWN high.

Please replace the paragraph beginning at page 79, line 1 through line 27 with the following amended paragraph:

An inhibit circuit (FIG. 26b), which includes a plurality of resistors 1110-1120, a diode 1122, a transistor 1124 and an FET 1126, is used to prevent the system from being turned ON during low battery conditions when the system is being supplied solely by the IBP 130. During a normal condition (i.e, when the system is being supplied power by the AC/DC converter or by the battery, the signal MAX786_VCC is connected to the main power signal POWER by way of the FET 1126. The FET 1126 is under the control of the transistor 1124. During conditions when the AC/DC converter is supplying power to the system, a signal AC/DCIN will be high, thereby turning ON the transistor 1124, which, in turn, turns ON the FET 1126, connecting the main power signal POWER to the signal MAX786_VCC. The collector of the transistor 1124, in turn, controls the FET 904, which connects the power supply signal HOOK_VCC to the enable terminals ON3 and ON5 on the power supply 851. When AC power is not available, the AC/DCIN goes low, leaving the control of the transistor 1124 under the control of an inhibit signal INHIBIT, available from the IBP 130 by way of the connector 850. During a normal battery condition, the inhibit signal is high, keeping the transistor 1124 turned ON, thereby enabling the power supply 851 by way of the FET 904. Should a low battery condition occur, the inhibit signal goes low, turning OFF the transistors 904, 1124, as well as the FET 1126, to prevent the system from being turned ON.

Please replace the paragraph beginning at page 82, line 4 through line 29 with the following amended paragraph:

The hot icons in the hot icon area 1202 (FIG. 36) are triggered by a pen-down event followed by a pen-up event. As discussed above, such a sequence of pen events is processed by hot icon ID processor 1210, illustrated in FIG. 32. The hot icon ID processor 1210 first determines if the pen event occurred in the viewing area 1200 (FIG. 36) of the LCD 113C by determining from the mouse mode handler 1214 (FIG. ~~[[33]]~~31) whether the system is in the TOUCH in step 1234, since this mode only occurs for pen events in the viewing area 1200 of the LCD display 113C. If the system is not in a TOUCH mode, the system checks in step 1236

whether the system is in the MOVE mode. If the pen event (i.e. pen-down followed by a pen-up event) did not occur in the viewing area 1200 of the LCD display 113C, the system compares the coordinates of the pen-down event with the locations of the various hot icons displayed in FIG. 37 in step 1238. In step 1240 (FIG. 32), the system determines if the left/right mouse button hot icon 1232 was selected. If not, the system proceeds directly to step 1242 to uplevel software for processing. If the system determines that the left/right mouse hot icon 1232 was selected, the system emulates a left or right mouse button in step 1244, depending on the last status of the left/right mouse button emulation and utilizes the emulated left or right mouse button status in the uplevel software in step 1242.

Please replace the paragraph beginning at page 82, line 30 through page 83, line 2 with the following amended paragraph:

Pen events in the hot icon area 1202 of the LCD display 113C are handled by the hot icon ID processor 1210 (FIG. [[32]]31), while pen events in the viewing area 1200 are handled by the mouse mode handler 1214 (FIG. [[33]]31). The mouse mode handler 1214 emulates two mouse actions: moving without either button being depressed and released (MOVE); and button depression and release events (TOUCH). As discussed above, both left and right mouse button events can be emulated in the TOUCH.

Please replace the paragraph beginning at page 83, line 3 through line 19 with the following amended paragraph:

As discussed above, a current pen-down event preceded by a pen-down event activates the mouse mode handler 1214 (FIG. [[33]]31). In step 1246, the system first determines if the hot icon flag is on. As discussed above, the hot icon flag is turned on anytime a pen-down event occurs in the hot icon area 1202 (FIG. 36) of the LCD display 113C. If the hot icon flag is not on, the pen-down event is translated to a mouse button down event by a mouse TOUCH handler in step 1248. If the hot icon flag is on, the system determines in step 1250 whether the coordinates of the current pen-down event to determine if the current pen-down event occurred

in the hot icon area 1202. If so, the pen coordinate data is dropped in step 1252 since such data will be processed by the hot icon ID processor 1210 (FIG. [[32]]31), discussed above. If the current pen event occurred in the viewing area 1200, the pen coordinate data is translated to mouse move data.

Please replace the paragraph beginning at page 87, line 31 through page 88, line 3 with the following amended paragraph:

The pen controller 110A (FIG. 21e) normally generates a series of interrupts and, in turn, a series of pen packets whenever the pen touches the LCD 113C (a pen-down event) and is lifted from the LCD 113C (a pen-up event) and generates an interrupt. For each interrupt, a single packet is generated. The format of the possible packets is illustrated in Table 7 below, where x0 is bit 0 of the x coordinate of the pen location and y0 is bit 0 of the y coordinate of the pen location, etc.

Please replace the paragraph beginning at page 94, line 14 through line 25 with the following amended paragraph:

If the protocol API call is not a register protocol stack or a bind stack service, the system checks in step 1396 whether send packet service is requested. If not, the system exits and the service call is handled by LSL.COM. If so, an IPXMUX send packet routine is called in step 1398 and 1400 (FIG. [[51]]50), which sets the address of the wireless LAN card 1360 as the source node address in step 1402. The packet modifier also sets the node address of the wireless interface device 100 as the packet's destination mode address in step 1400 (FIG. [[51]]50). The send event service routine address is set to the address of the send event service routine in IPXMUX.COM before it returns.

Please replace the paragraph beginning at page 97, line 9 through line 17 with the following amended paragraph:

In order to obviate the need to reconfigure the wireless interface device 100 the next time the wireless interface device 100 is connected to the host computer 101, the system checks in

step 1434 whether any of the configuration data (i.e. contrast, brightness (FIG. 57) was changed. If not, the wireless interface device 100 is placed in a suspend mode in step 1436. If the configuration data did change, the new configuration data is saved in the EEPROM 111B (FIG. 12h) in step 1438.

Please replace the paragraph beginning at page 99, line 24 through page 100, line 2 with the following amended paragraph:

Referring to FIG. 60, the system checks in step 1476 whether there has been a pen-down event in the viewing area 1202 (FIG. 36) of the LCD 113C and, in step 1478, whether the OSK hot icon 1480 (FIG. 37) was selected. If not, the system loops back to step 1476. If so, the selected key on the OSK is translated into a keyboard scan code in step 1480 and a visual indication of the key selected in the edit field 1472 or 1474 in step 1482. The process is repeated until the macro (i.e. WIN, DIR), followed by a carriage return 1486, is complete and the macro is saved in the EEPROM 111B (FIG. 12h) in step 1484. A clear button 1486 is provided in the dialog box illustrated in FIG. 54 for each edit field 1472 and 1474. These clear buttons 1486 enable the edit fields to be cleared in the EEPROM 111B (FIG. 12h).

Please replace the paragraph beginning at page 100, line 3 through line 18 with the following amended paragraph:

Activation of the remote keyboard macros is accomplished by pressing down on the user-defined hot icons 1472 or 1474, located in the hot icon area 1202 (FIG. 36) of the LCD 113C. The system checks in steps 1486 and 1488, whether the user defined hot icons 1472 or 1474 are selected. If the user-defined hot icons 1472 or 1474 are not selected, the system returns to step 1486. Once one of the user-defined hot icons 1472 or 1474 is selected, the keyboard scan code sequence, stored in the EEPROM 111B (FIG. 12h), is retrieved for the selected hot icon 1472 or 1474 in step 1490, which are then individually transmitted to the host 101 in step 1492. These scan codes are then written to the keyboard buffer on the host 101 in step 1494. Subsequently, in

step 1496, the host 101 processes the scan codes as though they originated from the host keyboard.

Please replace the paragraph beginning at page 100, line 20 through line 27 with the following amended paragraph:

As mentioned above, the wireless interface device 100 includes several flash memory devices 742-748 (~~FIG. 25~~FIGS. 25a-25c). The flash memory device 742 includes a protected area which contains the system BIOS, and a sufficient amount of functionality to enable the wireless interface device 100 to be rebooted to enable reprogramming of the flash memory devices 742-748 by way of the serial port 788 (FIG. 23b) in the event of a flash disaster.

Please replace the paragraph beginning at page 100, line 28 through page 101, line 2 with the following amended paragraph:

In order to upgrade the flash memory devices 742-748, the upgrade disks are installed in an available host computer 101. In particular, the flash upgrade software is written to a predetermined directory on the host's 101 hard disk. After the flash upgrade disks are installed, the wireless interface device 100 is turned on in step 1498 (FIG. 62A) by way of the main power switch 855 (FIG. 28a). Subsequently, in step 1500, a connection between the host computer 101 and wireless interface device 100 is initiated in step 1500 by first selecting the configuration hot icon 1410 (FIG. 37).

Please replace the paragraph beginning at page 101, line 3 through line 32 with the following amended paragraph:

Subsequently, the maintenance button on the dialog box is selected to get to the dialog box illustrated in FIG. 55. An upgrade button 1502 on the dialog box illustrated in FIG. 55 is selected in step 1504. In order to prevent programming errors, the radio quality is checked in step 1506 before proceeding. If the radio quality is poor, the upgrade is aborted. If the radio quality is adequate, power management is disabled in step 1508 to prevent the wireless interface device 100 from going into a reduced power state as discussed above during programming of the

flash memory devices 742-748. After the power management is disabled, a portion of the DRAM memory 111A (~~FIGS. 18 and 24~~FIG. 18a) in the wireless interface device 100 is set aside to receive a flash sector from the host computer 101 in step 1510. Subsequently, the wireless interface device 100 polls the host computer 101 to determine the correct numbers of sectors in the flash update and whether the sectors are available on the hard disk of the host computer 101 in steps 1512 and 1514. If the flash update files are not on the host hard drive or an incorrect number of sectors are available on the host hard disk, the update is aborted. Otherwise, the system requests the path/file data from the host computer 101 in step 1516. Subsequently, each sector (file) in the flash update is read by the host computer 101 and uploaded over the radio to the DRAM 111A in the wireless interface device 100 in step 1518. After the sectors are written to the DRAM 111A in the wireless interface device 100, a BIOS call is made to write the sectors in the DRAM 111A to the flash memory devices 742-748 in step 1520.

Please replace the paragraph beginning at page 102, line 31 through page 104, line 12 with the following amended paragraph:

As mentioned above, the wireless interface device 100 can be connected to any of the available hosts 101 that appear in the dialog box illustrated in FIG. 53 in the manner described above. The system illustrated in FIGS. 64A and 64B obviates the need for the user to select a host 101 for connection each time the wireless interface device 100 is powered up, by automatically connecting the wireless interface device 100 to the last host 101 to which it was successfully connected. As will be discussed in more detail below, when a host 101 is selected from the dialog box illustrated in FIG. 53 for connection to the wireless interface device 100 and a connection is successfully achieved, the node address of that host 101 is stored in the EEPROM 111B (FIG. 12h). Subsequently, once the wireless interface device 100 is powered up in step 1544, the system reads the node address from the EEPROM 111B, and reads it to a specific location in DRAM 111A (~~FIGS. 18 and 24~~FIG. 18a) in step 1546. After the node

address is written to the DRAM 111A, the system checks the node address to determine whether it is valid in step 1548. Invalid node addresses occur anytime the wireless interface device 100 makes an attempt to connect to a host 101, which fails during automatic reconnecting or is later disconnected by the end user. Thus, if a successful connection is not made or if there is a manual disconnection, the node address is cleared from the DRAM 111A in step 1550 and thus will be invalid. Subsequently, if the automatic reconnect fails in order to facilitate connection of the wireless interface device 100 to another available host 101, the set-up dialog box illustrated in FIG. 53 is displayed on the display 113C of the wireless interface device 100 in step 1552. After the host selection set-up dialog box is displayed on the wireless interface device 100, the system checks in step 1554 whether the wireless interface device 100 is connected to an available host 101. Normally, if an invalid address is found in step 1548 and the host selection set-up dialog box appears on the display 113C of the wireless interface device 100, there will be no connection to an available host 101 and the system will jump to step 1556, where it checks if the hot icon area 1202 (FIG. [[37]]36) has been depressed. Normally in this situation, since the host selection dialog box is already being displayed on the screen 113C of the wireless interface device 100, the only hot icon that can affect the situation is a sleep-face hot icon 1558 (FIG. 37), which places the wireless interface device 100 in a low-power sleep mode. In a normal situation when the wireless interface device 100 is first powered up, the sleep-faced hot icon 1558 is not depressed and the system waits for the user to select an available host 101 from the host set-up dialog box illustrated in FIG. 53 as discussed above in step 1560. Once an available host 101 is selected, the system loops back to step 1562 and attempts to establish connection with the selected host 101.

Please replace the paragraph beginning at page 104, line 13 through line 24 with the following amended paragraph:

In step 1564 the system checks whether or not the connection was successful. If not, the system goes to step 1550 and clears the node address for the selected host 101 from the DRAM

111A and displays the host selection set-up dialog box in step 1552. If the connection between the wireless interface device 100 and the host 101 is successful, the node address of the host 101 is saved in a specific DRAM location in step 1566, and in turn, written to the EEPROM 111B (FIG. 12h) in step 1568. After the node address of the selected host 101 is stored in EEPROM 111B, the wireless interface device 100 will display whatever is being displayed on the host 101 in step 1570.

Please replace the paragraph beginning at page 105, line 20 through page 106, line 6 with the following amended paragraph:

The system discussed above is thus able to automatically connect the wireless interface device 100 to the last host 101 to which it was connected. After the automatic reconnect, should the set-up window hot icon 1410 (FIG. 37) be selected, the host selection set-up dialog box illustrated in FIG. 53 will be displayed on the screen 113C of the wireless interface device 100. Subsequently, the system will go to step 1554 and check whether the wireless interface device 100 is connected to a host 101. In this case, since the wireless interface device 100 will still be connected to the available host, the system then checks in step 1586 whether a disconnect button on the host selection dialog box illustrated in FIG. 53 has been depressed. If not, the system goes to step 1556 and continuously waits for a hot icon in the hot icon area 1202 (FIG. 36) of the LCD 113C to be depressed. If the disconnect button in the host selection set-up dialog box illustrated in FIG. 53 is depressed, the node address for the host 101 to which the wireless interface device 100 is connected is erased from the specific location in the DRAM 111B in step 1588. Subsequently, the system goes to step 1556 and waits for a hot icon in the hot icon area 1202 (FIG. [[37]]36) to be depressed.

Please replace the paragraph beginning at page 106, line 8 through line 23 with the following amended paragraph:

As mentioned above, the wireless interface device 100 includes a virtual on-screen keyboard (OSK), as illustrated in ~~FIG. 66~~FIGS. 66a and 66b. More particularly, the OSK is

configurable by the buttons 1590, 1592, 1594 and 1596 in a control box located at the top of the OSK. These buttons 1590, 1592, 1594 and 1596 enable the OSK to be configured. For example, a button 1590 displays the OSK as illustrated in FIG. 66A with a full keyboard and numeric keypad. The button 1592 is a toggle which displays the keyboard without the numeric keyboard as illustrated in FIG. 66B. The button 1594 displays the numeric keypad with the NUM LOCK off as illustrated in FIG. 66C, or alternatively displays the OSK as a numeric keyboard NUM LOCK on as illustrated in FIG. 66D. The button 1596 allows the size of the OSK to be varied. The "X" button closes the window displaying the OSK.

Please replace the paragraph beginning at page 109, line 16 through page 110, line 6 with the following amended paragraph:

The alternate embodiments of the invention discussed heretofore all relate to a single wireless interface device 100 interfaced to a single host implemented as a personal computer or to a local area network by way of an access point 109. The following embodiments illustrated in FIGs. 67-[[112]]85 primarily relate to a system in which multiple wireless interface devices 100 interface in real time with a multi-device server which forms a portion of either a wired LAN or a wireless LAN, or multiple servers connected together by routers, as will be discussed in more detail below. The system for enabling multiple wireless interface devices 100 to interface in real time with a multi-device server or plurality of servers is generally identified with the reference numeral 1700 and illustrated in FIG. 67. In this system 1700, a plurality of wireless interface devices 100a, 100b, 100c, 100d, etc. communicate with one or more local area network (LAN) segments 1702 and 1704, by way of an access point 109 (discussed above). Each LAN segment 1702, 1704 includes a multi-device server 1708, 1710 with an extended Windows NT operating system, as discussed below. The LAN segments 1702 and 1704 are connected together by a router 1706, discussed in more detail below. Only four wireless interface devices, identified in FIG. 67 as 100a, 100b, 100c and 100d, are shown for example. However, more wireless interface devices 100 can be connected to the System 1700.

Please replace the paragraph beginning at page 112, line 24 through line 34 with the following amended paragraph:

The system for enabling wireless enumeration of the servers available for connection to a wireless interface device 100 is illustrated in ~~FIGs. 67-74~~FIGs. 67-70, 71a-71c, 72-74. FIG. 67 is an overall flow chart for both the servers 1708, 1710 and wireless interface devices 100. The software for the wireless interface device 100 is illustrated in FIGs. 71a-71c, while the server software is illustrated in FIGs. 72-74. FIG. 70 illustrates a set-up dialog box, available at the wireless interface device 100 for initiating the wireless enumeration of the servers and connecting to one of the servers.

Please replace the paragraph beginning at page 120, line 10 through line 25 with the following amended paragraph:

In order to minimize memory storage space, local software for the wireless interface device 100 is stored in a compressed format, for example, in a read only memory device (ROM), such as the flash memory devices 742-748 (~~FIG. 25~~FIGS. 25a-25c), then decompressed, written and executed from the DRAM memory devices 111A (FIG. 18a). As will be discussed in more detail below, both .EXE files and .COM files, as well as various other types of files are compressed and decompressed. An .EXE file is any executable file with an extension .EXE, i.e., FIND.EXE, MSD.EXE. A .COM file is any executable file with an extension .COM, i.e., EDIT.COM, SYS.COM. Such files, as known by those of ordinary skill in the art, include a header portion as well as a data, or code portion, where either data or a software program is stored. An exemplary header for an .EXE file is illustrated in Table 8 below.

Please replace the paragraph beginning at page 122, line 4 through line 18 with the following amended paragraph:

For applications to be run locally on the wireless interface device 100, which include a number of .EXE files, the header for such an .EXE file can occupy a relatively substantial portion of the available memory storage space provided by the flash memory devices 742-748

(~~FIG. 25~~FIGS. 25a-25c). In order to reduce the required memory storage space in the flash memory devices 742-748 in the wireless interface device 100, the headers for the .EXE files are at least partially compressed, in accordance with an important aspect of the invention. As will be discussed in more detail below, the header for such .EXE file is transformed into a customized header 1882 (FIG. 79), which may include an uncompressed portion 1884 and a compressed portion 1886. The data or code portion 1888 is totally compressed, as discussed above.

Please replace the paragraph beginning at page 123, line 3 through line 17 with the following amended paragraph:

New software to be loaded into the wireless interface device 100 may be loaded by way of the UART 788 (FIG. 23b) by way of the serial port 790 (FIG. 30a), or by way of the radio interface 960 (FIG. 16a). In particular, in order to load software into the wireless interface device 100 wirelessly in a system in which multiple wireless interface devices 100 are supported by a single server, the software is first loaded into an available server 1708, 1710 (FIG. 67). In such an application, the wireless interface device 100 is placed in a set-up mode of operation. In particular, the hot icon 1410 (FIG. 37) is initially selected from the hot icons 1202 (FIG. 36) illustrated in FIG. 55. The MAINTENANCE BUTTON 1392 is then depressed to provide the dialog box as illustrated in FIG. 55.

Please replace the paragraph beginning at page 123, line 18 through page 124, line 14 with the following amended paragraph:

The overall flow chart for the compression/decompression process is shown in FIG. 75. Initially, files are compressed and transmitted to the wireless interface device 100. In particular, the compressed files are written directly to the flash memory devices 742. In order to execute the file, the compressed file from the flash memory device 742 is written to a temporary file within the DRAM memory devices 111[[a]]A (FIG. 18a) in the memory space 10000 to 1FFFFF. In such an application, the flash memory devices 742 act as input files, while the temporary file in the DRAM memory devices 111[[a]]A serves as an output file. Alternatively, new files to be

written to the flash memory devices 742 are initially uncompressed and stored in an external input file 1896, external from said wireless interface device 100. The input file 1896 is then compressed and stored in an output file 1898. The compressed output file 1898 is then transferred to the flash memory devices 742 within the wireless interface device 100 over a radio link. Thus, in step 1900, depending upon whether compressed data is being written to the flash memory devices 742, or whether the compressed data within the flash memory device is being executed, input and/or output files 1896, 1898 are opened in step 1900 as generally discussed above. If the file is to be transferred to the flash memory devices 742 in the wireless interface device, the file is compressed and written to an output file 1898 and transferred to the flash memory devices 742, as indicated by steps 1902 and 1904. For files that are currently stored in the flash memory devices 742 in a compressed format, these files are decompressed and written to an output file 1898 for execution as indicated in steps 1902 and 1904.

Please replace the paragraph beginning at page 124, line 15 through page 125, line 2 with the following amended paragraph:

The software for compressing the various software to be stored in the wireless interface device 100 is illustrated in FIGs. 76a and 76b. Files to be compressed are read to determine whether the file is an .EXE file or a .COM file in step 1910. The system then sets up a signature field 1890 (FIG. 79). As discussed above, the signature field 1890 is stored in the uncompressed portion 1884 of the customized header 1882 and may include information as to whether the file is an .EXE file or a .COM. Thus, in step 1910, the input file 1396 is read to determine the type of file written to the input file 1896 (FIG. 80b). If the file is an .EXE file, a signature flag for an .EXE file is set in the signature field 1390, as illustrated in step 1912. On the other hand, if the file is a .COM file, the signature flag within the signature field 1890 (FIG. [[70]]79) is set to represent a .COM file in step 1914. Once the signature flag is set, other signature information may be added to the signature field 1890 in step 1916. For example, as discussed above, the software version of the compression software may be included in the signature field 1890 in

order to speed up the decompression process. Once the signature field is set up, the signature field is written to the output file 1898 in step 1918.

Please replace the paragraph beginning at page 127, line 18 through page 128, line 23 with the following amended paragraph:

The flow chart for decompressing stored compressed files in the flash memory devices 742-748 is illustrated in ~~FIG. 77~~FIGS. 25a-25c. Initially, any file to be executed is in a compressed format as discussed above. Initially, as indicated by step 1946, the signature field 1890 (FIG. [[78]]79) is read from the input file 1896. After the signature field 1890 is read from the input file 189 6, the customized file header 1882 is read in step 1948. As mentioned above, the signature field 1890 identifies whether the particular file is an .EXE file or a .COM file. Thus, the system ascertains in step 1950 whether the file is an .EXE file or a .COM file. As indicated above, the signature field 1890 (FIG. 79) may include data regarding the file as to whether it is an .EXE file or a .COM file, as well as the software version of the compression software in order to speed up the decompression process. Before the file can be decompressed, the size of the compressed data or code portion 1888 (FIG. 79) must be ascertained. As indicated above, for .EXE files, the size of the header may be ascertained directly from the customized file header 1882 (FIG. 79). Since the header for a .COM file is compressed in the same manner as the code portion 1888 for the .COM file, the header portion 1882 is treated the same as the code portion 1888. Thus, the entire .COM file, header portion 1882 and code portion 1888 are written directly into the output file 1898 (FIG. 78) in step 1952. In the case of .EXE files, the customized file header 1882 is written to the output file 1898. The system then reads the size of the block in step 1954. In the case of a .COM file, the size of the compressed data or code block may be read directly from the flash memory device 742. In the case of an .EXE file, the file header is partially compressed, as indicated above, in data blocks. Thus, in steps 1954-1958, the system reads decompressed blocks of data from the input file 1896 and writes the decompressed data to the output file 1898. Both the headers portions 1882, as well as

the data or code portions 1888 are decompressed one data block at a time by the loop consisting of the steps 1954-1958. Once all of the data has been decompressed, including the header, the decompressed file may be executed directly from the output file 1898, which may be a part of the DRAM 111A.

Please replace the paragraph beginning at page 128, line 25 through page 129, line 15 with the following amended paragraph:

As previously indicated, the wireless interface devices 100 may include one or more flash memory devices 742-748 (~~FIG. 25~~FIGS. 25a-25c). However, the present invention also applies to other electronically programmable memory storage devices, such as electronically erasable programmable read only memory (EEPROM). For a "single user" system, as indicated above, any software updates to the wireless interface device 100 may be accomplished by loading the software onto an available host 101 and then establishing a connection between a host computer 101 and the wireless interface device 100. For a "single user" wireless interface device, as discussed above, the user simply goes to the set-up dialog box, as indicated in FIG. 55, and depresses the upgrade button for automatic, wireless loading of the software to the wireless interface device 100. In a multi-user environment, for example, as illustrated in FIG. 67, each of the wireless interface devices 100 can individually initiate an upgrade from the available server 1708, 1710. In such an application, the server, and, in particular the network administrator notifies all of the various wireless interface devices 100a-100d users connected to the network 1700 that the local software within the wireless interface device needs to be updated. Each of the individual wireless interface devices 100 can then be updated from the server 1708 wirelessly, as illustrated in FIGs. 80-85 and discussed below.

Please replace the paragraph beginning at page 129, line 16 through page 131, line 2 with the following amended paragraph:

More particularly, initially, each of the wireless interface devices 100a-100d are turned on in step 1960 (FIG. 80a) and a connection is established with the system servers 1708 in

step 1962 as discussed above. Once the connection with the server 1708 is established, each of the individual wireless interface devices 100a-100d is notified by the network administrator regarding the need for a local software update. The software in each of the individual wireless interface devices 100a-100d can be initiated by a local user interface as indicated in step 1964. In particular, the flash upgrade is initiated by going to the set-up dialog box on the wireless interface device 100 and depressing the MAINTENANCE BUTTON to arrive at the dialog box as indicated in FIG. 55. The user depresses the upgrade button to initiate automatic wireless installation of the software into the flash memory devices 742-748 in the wireless interface device. In order to prevent programming errors, the radio quality is checked in step 1966 before proceeding. If the radio link quality is poor, the upgrade is aborted, as indicated by step 1968. If the radio link quality is sufficient, any power management functions in the wireless interface device 100 is disabled in step 1970 to prevent the wireless interface device 100 from going into a reduced power state, as discussed above, during programming of the flash memory devices 742-748. After the power management function is disabled, a portion of the DRAM memory 111A (FIGs. ~~18-24~~18a-18d, 19a-19r, 20a-20c, 21a-21c, 22a-22d, 23a-23e, 24a-24b) in the wireless interface device 100 is set aside to receive a flash sector from the server 1708 (FIG. 67) in step 1972. Subsequently, in step 1974, wireless interface device 100 polls the servers 1708, 1710 to determine the total number of sectors in the flash update over the radio link in step 1974. After the total number of sectors is obtained for the upgrade, the system checks in step 1976 whether there have been any errors in the data transmission from the server 1708. The errors may be checked, for example, by checking whether cyclic redundancy checking (CRC) code matches a specified CRC code for each file or whether there are any other server errors. Thus, if the value resulting from the CRC at the wireless interface device 100 does not match the value of the CRC of the server 1708, the flash upgrade is aborted in step 1968. Otherwise, the system proceeds to step 1978 and sets up a receiving buffer in the DRAM memory devices 111A and requests a sector of the upgrade from the server 1708. Once a request for a sector is initiated, the

sector is transmitted from the servers 1708, 1710 over the radio link to the DRAM memory devices 111A. A BIOS routine is called in step 1982 to write the flash sector from the DRAM memory device 111A to the flash memory devices 742-748 in step 1982. In step 1984, the system checks for any errors in writing the flash sectors to the flash memory devices 742-748. Should any errors be detected, the flash upgrade is aborted and the system returns to step 1968. If no errors are detected, the system checks in step 1986 whether additional sectors need to be requested from the server 1708. If so, the system loops back to step 1978. If all of the sectors have been requested, the system goes to step 1988 and reboots the wireless interface device 100.

Please replace the paragraph beginning at page 134, line 26 through page 135, line 9 with the following amended paragraph:

FIGs. ~~86-89~~86-88, 89a-89b relate to the audio input processing system 2047. The audio input processing system 2047 includes an input path which, in turn, is connected to the microphone 2046 and an outpath which is connected to a speaker 2045. Audio input data is received by the microphone 2046 and filtered, for example, by a low-pass filter 2047 selected to pass signals of 3 Khz or less to only permit data in the voice range to be amplified by an amplifier 2049 and converted to a digital signal by an A-D converter 2051. The amplifier 2049 is used to increase the amplitude of the signal to produce a voltage reference to maximize the range of the analog-to-digital converter 251. The output of the A-D converter 2051 may be applied directly to the data bus, as discussed above, or may be applied to a digital signal processor 2053, for example, a Model No. CS 4237B or CS 4236B, as manufactured by Crystal Semiconductor; a Model No. ES-5510, as manufactured by Ensonic; or a Model No. SAA7710T, as manufactured by Phillip Semiconductor, all preloaded with factory installed firmware.

Please replace the paragraph beginning at page 137, line 25 through page 138, line 17 with the following amended paragraph:

FIG. 90 relates to registering an occlusion window on the servers 1708, 1710. In particular, an occlusion window is registered to prevent the server from overwriting the OSK on

the wireless interface device 100. The occlusion window, for example, in a Windows NT server, relates to a no-paint window within a portion of the viewing area 1260 (FIG. [[36]]34) of the LCD 113c for the wireless interface device 100. The occlusion window corresponds to the window displayed on the wireless interface device 100 for the on-screen keyboard (OSK). For each window in a network system, the window is registered with the server window system in step 2076. The occlusion window is registered by registering the class of the window, as indicated in step 2078 by calling the Register Class API. The class of the window refers to the various attributes of the window, for example, a dialog box or no-paint window. Once the class of the window is registered with the server window system, in order to make the occlusion window visible to all windows running on the system 1708, 1710 at one time, memory space in the servers 1708, 1710 is created for the occlusion window global data in step 2080. The global data relates to the position and dimensions of the on-screen keyboard. Thus, the OSK can be utilized on the wireless interface device 100 during conditions when multiple windows are running and even overlapping windows, as indicated in FIG. 95. As such, the OSK program may be formulated as a dynamic link library (DLL) that can be used by any windows running in the system.

Please replace the paragraph beginning at page 150, line 10 through line 32 with the following amended paragraph:

FIGs. 108 and 109a-109b illustrate the software at the wireless interface device for processing pen points. All points touched by the pen are stored in a buffer. Initially, the wireless interface device 100 powers up in a mouse mode and interprets all pen down events as mouse left button down points and assembled into data packets. Once it is determined that the system is in a pen mode, for example, when a pen down event occurs within an ink field 1642, the pen data points are assembled into pen data points and stored in a pen data buffer in the wireless interface device 100 and wirelessly transmitted to the server 1708, 1710, and, in particular, to the pen data buffer manager and the pen data processor at the servers 1708, 1710. As indicated in step 2262

(FIG. 108), pen down and pen up events are assembled into pen data packets and stored in a pen data buffer in the wireless interface device 100. After each point is stored in the pen data buffer, the point is sent to a router module for processing. Thus, after a pen data packet is assembled, the system checks in step 2264 to determine whether the router is busy. If so, this module will return. If the router is not busy, the router is called in step 2266 to process the pen data point.

Please replace the paragraph beginning at page 150, line 33 through page 151, line 11 with the following amended paragraph:

The flow chart for the router is illustrated in FIG. 109a. Initially, in step 2268, the system determines whether the wireless interface device 100 is in an ink mode, as discussed above. If the wireless interface device 100 is not in an ink mode, the system assumes that the wireless interface device 100 is in a mouse mode and proceeds to get the packet for the point from the data buffer in step 2270. This point is pushed onto the router stack in step 2272. The mouse manager is called in step 2274 to process the point as a mouse data point, as discussed above. The system continuously processes the points in the buffer while the system is in a mouse mode, until it is determined in step 2276 that the buffer is empty, at which point the system exits the router in step 2278.